l + special + special + special

SONDERHEFT NR. 2/85
DM 14,80/ÖS 124/SFR 14,80

Anwender-Programme
Utilities
Schul-Programme
Spiele
Adventures

Rund 150 Seiten Listings für den 99/4A

SPIEL-SALON
FÜR BUCHHALTER
MUSIKER & SAMMLER
MATHE-TABELLEN
TORE
DIE HANDWERKER KOMMEN
FÜR JEDEN ETWAS
WAS MAN SCHWARZ AUF WEISS BESITZT
SPRITE-KONSTRUKTEUR

Das Magazin für TI-99

special + special + special + s

Verlag und Redaktion haben eine neue Telefonnummer!
089/1298011 für Verlag
089/1298013 für Redaktion.

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite
ACHTUNG!!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33220 und für Listings/Programme: 089 / 129 80 13

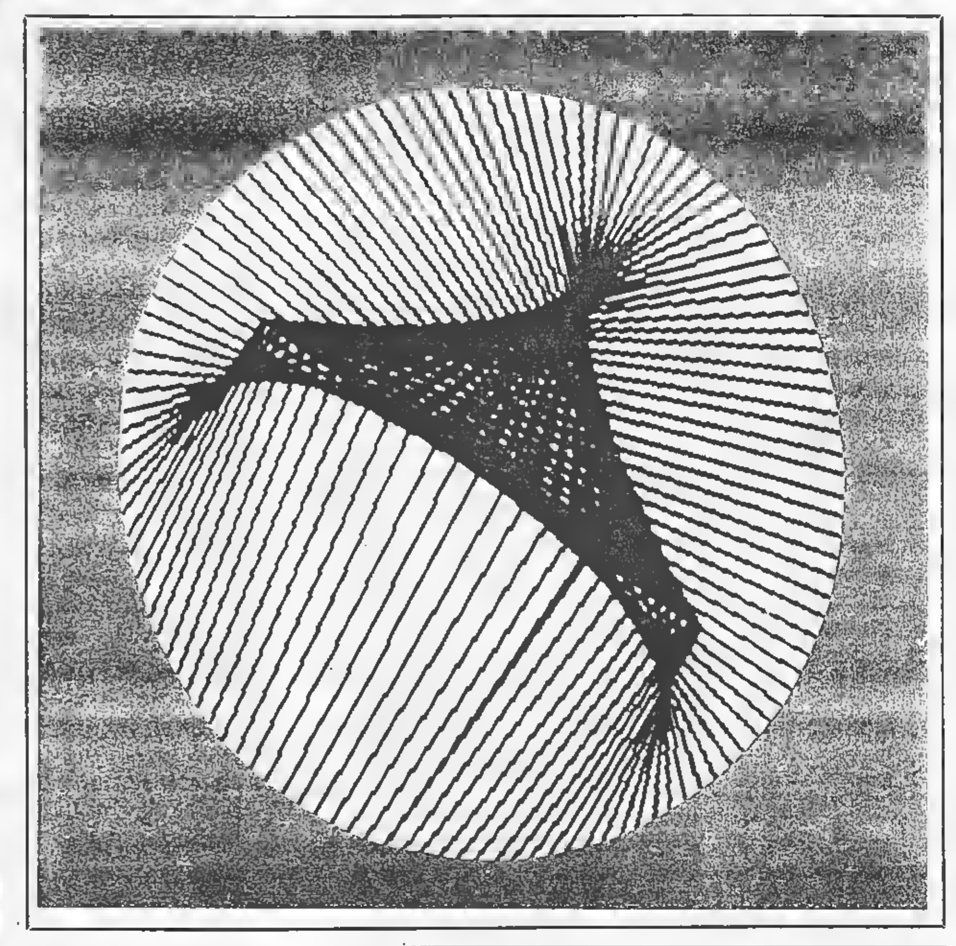
SONDERHEFT

Nr. 2/85

INHALT

UTILITIES SCHULE ANWENDER SPIELE IN TI BASIC

UTILITIES SCHULE ANWENDER SPIELE IN EX BASIC



Impressum

TI-SPECIAL ist eine Sonderausgabe der TI-REVUE. Sie erscheint im TI/C8M-Verlag, Elisabethstraße 1, 8044 Lohhof.
Redaktion: Senator-Presse-Service.
Verantwortlich für den Inhalt: Hartmut König, Werner E. Seibt.
Verantwortlich für Anzeigen: Bruno Redase. Alle: Postfach 1107, 8044 Lohhof. Anfragen bitte nur schriftlich.
Vertrieb: Verlagsunion Wiesbaden.
Es gilt die Honorarliste des Verlages. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung. Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck und die Aufnahme in den Kassetten-Service.

Erstverkaufstag dieser Ausgabe: 27. September 1985
Es gilt Preisliste Nr. 5 vom 1.1.1985.
Media-Unterlagen bitte anfordern bei Bruno Redase, co TI/CBM Verlag, Postfach 1107, 8044 Lohhof.
Für telefonische Anfragen: 089/129 80 11
Printed in Germany by Mayer & Söhne

DER DEBUGGER UND SEINE MÖGLICHKEITEN

Aufgrund der Tatsache, daß die Zahl der Editor/Assembler-Besitzer laufend steigt und somit ebenfalls (hoffentlich) der Assembler Programmierer, habe ich mich entschlossen, Teile des Debuggers näher zu erläutern.
Nach genauem Studium, sowohl der englischen als auch der deutschen Beschreibung, zeigten sich einige Mängel, die ich hiermit beheben möchte. Der erste Befehl auf den ich näher eingehen möchte, gestattet das Auslesen des Assemblermoduls, sowie das Inspizieren sämtlicher Grom-Module.
Mit dem Kommando M G6000,7800 zeigt sich der Inhalt des E/A-Moduls, das nach dem Laden des Debuggers auch aus dem Gromport entfernt werden kann. Jetzt könnte z.B. das Statistik-Modul ins Port gesteckt und ausgelesen werden, wenn nicht beim Einschub eines Moduls ein Resetimpuls auftreten würde.
Dieser Reset kann auf einfache Weise bei Modulen, die nur mit Groms (Graphic Read Only Memory) bestückt sind, außer Kraft gesetzt werden. Die nachfolgende Liste führt einige dieser Module auf:
Ti-Writer
Text- und Dateiverwaltung
Editor/Assembler
Datenverwaltung und Analyse
Statistic
Multiplan
Buchungsjournal
Wumpus
Speech-Editor
Amazing
Adventure usw.

Sollten Sie im Zweifel darüber sein, ob Ihr Modul ein reines Grom-Modul ist, genügt das Öffnen des Gehäuses.
Treten dabei nur ICs zutage mit maximal 16 Anschlüssen, so besteht kein Zweifel, daß es sich um ein nur mit Groms bestücktes Modul handelt. Zum Auslesen des Moduls genügt das Überkleben des Pins 1 mit einem Stück Tesafilm. Dieser Pin befindet sich, wenn Sie Ihr Modul von vorne betrachten (Steckerseite, die in den Gromschlitz gesteckt wird), rechts unten. Nach dieser Behandlung können Sie Ihr Modul jetzt in das Grom-Port stecken und auslesen.

MODIFIKATION OHNE EINGRIFFE

Diese Modifikation kann ohne irgendwelche Eingriffe ins Modul auch von außen durchgeführt werden.
Die nachfolgende Tabelle zeigt die gültigen Adreßbereiche der Groms:

M G6000,7800	Grom 1
M G8000,9800	Grom 2
M GA000,B800	Grom 3
M GC000,D800	Grom 4
M GE000,F800	Grom 5

Noch ein Tip zum Auslesen.
Leider hat sich gezeigt, daß manchmal trotz „Spezialbehandlung" ein Reset auftritt. Dieser Fehler kann durch folgende Punkte weitgehend eingedämmt werden.

1. Start des Debuggers
2. Befehl M 0,FFFF
3. Entfernen des E/A-Moduls
4. Einstecken des auszulesenden Moduls
5. Anhalten der Auflistung mit FCTN X
6. Auslesen laut obiger Tabelle
7. Bei Absturz Neubeginn mit Punkt 1

DSR AUSLESEN

Der zweite Befehl, auf den ich näher eingehen möchte, bezieht sich auf das Auslesen der Peripherie-Roms (DSR).
Die meisten Karten, u.a. P-Code, RS232, Controller, beinhalten ein Rom mit ihrem Betriebssystem. Diese Karten liegen auf folgenden CRU-Adressen.

1100	Controller
1300	RS232 I
1500	RS232 II
1800	Thermalprinter
1F00	P-Code

Der Adreßbereich des DSR-Roms befindet sich bei 4000 – 5FFF. Je nach Notwendigkeit wird vom System eines der Peripheriegeräte zu- oder abgeschaltet. Dieses Zuschalten bei Gebrauch bietet gegegenüber anderen Computersystemen den Vorteil, daß bei einer Peripherieerweiterung kein Rambereich verloren geht.
Wie kann man nun z.B. das Betriebssystem der RS232 I auslesen?
Mit dem Kommando C 1300,1 <Enter> erfolgt die Ausgabe C 1300,XXXX die mit einer Eingabe von 1 <Enter> abgeschlossen wird. Nun können Sie das Rom im Bereich 4000–5FFF auslesen.

AUCH FÜR MINIMEMORY

Der Befehl C 1300,1 bewirkt dabei, daß nur ein Bit ab der Cru-Adresse 1300 Hex ausgegeben werden kann. C 1300,0 würde eine 16 Bit breite Ausgabe ab 1300 erlauben.
Bevor Sie nun ein anderes Rom auslesen, muß erst das Cru-Bit bei 1300 zurückgesetzt werden. Das Rücksetzen erfolgt mit:

	C 1300,1<Enter>
Ausgabe	C 1300,XXXX
Eingabe	0 <Enter>

Sollte jemand seine spielerische Ader mit dem Einbzw. Ausschalten der Peripherie-Leuchtdioden befriedigen wollen, so kann er dies durch das Senden eines Bits auf folgenden Adressen:

1100	/1100	Controller
130E	/1300	RS232 I
150E	/1500	RS232 II
1F00	/1F00	P-Code
---	/ -------	
Led ein bzw. aus	/ Bit setzen zum Auslesen der DSR (DSR=Device Service Routine).	

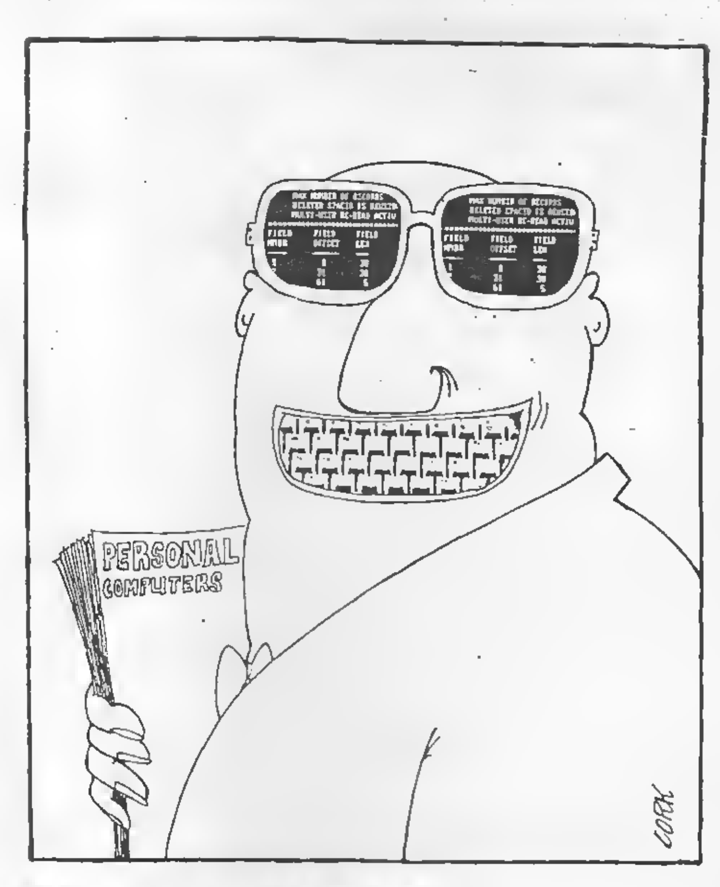
Aus dieser Tabelle wird erkennbar, daß sowohl bei der Controller- als auch der P-Code-Karte nur ein Bit für das Aufleuchten der Led und das gleichzeitige Auslesen gesetzt werden muß.
Zum Schluß möchte ich noch anmerken, daß auch Minimemory-Besitzer den Debugger starten und somit verwenden können. Eine Kopie von einem Freund oder Bekannten wird sich bestimmt als nützlich erweisen.

Reiner Ziegler

ACHTUNG!!!

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite
ACHTUNG !!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13



Space 2000

Der Wasserplanet ‚Atlantis' wird von einer Flotte feindlicher Raumschiffe angegriffen. 20 Gegner trachten nacheinander, den Planeten zu zerstören. Sie senken sich immer weiter hinab, bis sie die Verteidigungsbasis erreicht haben.

Sie, als joystickgewappneter Spieler, müssen versuchen, dies zu verhindern. Mit dem Fadenkreuz visieren Sie den Gegner an und zerstören ihn durch Betätigen des Feuerknopfes. Zu Beginn des Spieles haben Sie tausend Energieeinheiten zur Verfügung. Jeder Schuß kostet zehn Einheiten. Können Sie sich vor den Angreifern nicht mehr retten und bauen mit der SPACE-Taste einen Schutzschirm um die Basis auf, so müssen Sie dafür 100 Energieeinheiten opfern.

Die Invastion ist abgewehrt, wenn keines der 20 angreifenden Raumschiffe Ihre Basis zerstören onnte. Sollte alle 1000 Energieeinheiten vorzeitig verbraucht werden, ist das Spielvorzeitig beendet.

Jörg Bublath

Komplexe Zahlen

Das Programm bietet folgende mathematischen Operationen mit komplexen Zahlen: Transformation von Polar- in kartesische Koordinaten und umgekehrt, Addition, Subtraktion, Multiplikation, Division; Bilden von Wurzel, Kehrwert und Konjungiert kom. Zahl. Außerdem besteht die Möglichkeit, Zwischenergebnisse zu speichern.

Die Wahl der jeweilgen math. Operationen und die Art der Eingabe (z.B. in Kart.- oder Polarkoordinaten) erfolgt durch Steuern eines Pfeiles in einem Menüfeld.

Gerhard Hübner

```
100 ! *********************
110 ! *                   *
120 ! *     SPACE-2000    *
130 ! *                   *
140 ! *    Copyright by   *
150 ! *                   *
160 ! *   Joerg  Bublath  *
170 ! *                   *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole *
210 ! *      Ext. Basic   *
220 ! *                   *
260 ! *  Speicherbelegung *
270 ! *     8544 Bytes    *
280 ! *                   *
290 ! *********************
300 DIM M$(5)
310 CALL CLEAR :: CALL SCREE
N(2)
320 CALL CHARSET :: CALL DEL
SPRITE(ALL)
330 FOR I=2 TO 8 :: CALL COL
OR(I,16,1):: NEXT I
340 EN=1000
350 SC=0
360 ! TITEL BILD (J/N)
370 QW=QW+1 :: IF QW>1 THEN
770
380 !---------------------
390 !DEFINITIONEN FUER
400 !     TITELBILD
410 !---------------------
420 CALL CHAR(40,"FFFFFFFF")
430 CALL CHAR(41,RPT$("0F",8
),33,RPT$("F",16))
440 !---------------------
450 ! BILDSCHIRM AUFBAUEN
460 !---------------------
470 DATA 3,4,7,4,11,4,3,10,6
,10,3,15,6,15,3,20,11,20,3,2
5,7,25,11,25
480 FOR I=1 TO 12 :: READ A
:: READ B :: CALL HCHAR(A,B,
33,4):: NEXT I
490 DATA 15,4,19,4,22,4,15,1
1,22,11,15,18,22,18,15,25,22
,25
500 FOR I=1 TO 9 :: READ A :
: READ B :: CALL HCHAR(A,B,3
3,5):: NEXT I
510 DATA 4,10,4,15,4,18,4,20
,4,25,15,11,15,15,15,18,15,2
2,15,25,15,29
520 FOR I=1 TO 11 :: READ A
:: READ B :: CALL VCHAR(A,B,
33,8):: NEXT I
530 DATA 4,4,7,7,16,8,19,4,3
,13
540 FOR I=1 TO 5 :: READ A :
: READ B :: CALL VCHAR(A,B,3
3,4):: NEXT I
```

```
550 CALL CHAR(64,"0000000000
000010"):: DISPLAY AT(2,4):"
SCYPSOFT PRESENTS :" :: DISP
LAY AT(23,3):"COPYRIGHT BY J
@ BUBLATH"
560 DISPLAY AT(13,4):"PRESS
ANY KEY TO BEGIN"
570 !---------------------
580 ! BLINKEN DER BANDE
590 !---------------------
600 CALL COLOR(1,16,1)
610 CALL HCHAR(1,2,41,30)::
CALL HCHAR(24,2,41,30):: CAL
L VCHAR(1,2,40,24):: CALL VC
HAR(1,31,40,24)
620 CALL KEY(0,K,S):: IF S=1
 THEN 690 ELSE CALL COLOR(2,
7,2)
630 FOR I=1 TO 40 :: NEXT I
640 CALL KEY(0,K,S):: IF S=1
 THEN 690 ELSE CALL COLOR(2,
2,7)
650 FOR I=1 TO 40 :: NEXT I
660 GOTO 620
670 !    ABFRAGE FUER
680 !  SPIELANLEITUNG
690 CALL CLEAR :: DISPLAY AT
(7,1):"ANLEITUNG?" :: ACCEPT
 AT(7,12)VALIDATE("JN")SIZE(
1):AN$ :: IF AN$="J" THEN 17
60
700 DISPLAY AT(7,1):"INITIAL
ISIEREN-BITTE WARTEN"

710 !---------------------
720 ! DEFINITION DER
730 ! RAUMSCHIFFE UND
740 ! DES BILDSCHIRM-
750 ! AUFBAUES
760 !---------------------
770 CALL CHAR(96,"208A005402
48220000AA0022801440128024O1
5004A0055020884200550OA208")
780 CALL CHAR(100,"000000000
100010B0B010001000000000000
000800080D0D080008")
790 CALL CHAR(104,"183830706
0606060",105,"181C0C0E060606
06",106,"03030C0C3030C0C0")
800 CALL CHAR(107,"C0C030300
C0C0303",108,"0303060606060C
0C",109,"C0C0606060603030")
810 M$(1)="000040FF41033B6FF
F6F3B0341FF400000003FE83C0C
0FDFEFDC0C083FE0300"
820 M$(2)="00000033F7FFF999
9FF7F3F03000000000000C0FCFEF
F9999FFFEFCC0000000"
830 M$(3)="00000000030F3FFFF
F3F0F030000000040E1F3EFCF8F
0E0E0F0F8FC3E1F0E04"
840 M$(4)="0000000003F63FFF
F633F0000000000000001C2241FCF
8F2F2F8FC41221C0000"
850 CALL CHAR(110,"000000000
0FFFF00",115,"000010381")
860 CALL CHAR(112,"000000181
8",113,"00000018",114,"00000
01")
870 A$="000000030F1F3FFF" ::
 B$="000000C0F0F8FCFF" :: CA
LL CHAR(118,"80C0E0F0F8FCFEF
F")
880 CALL CHAR(33,A$,34,B$)
890 CALL CHAR(119,"0103070F1
F3F7FFF",120,"8442211008854 2
24244285081021428421428408 10
A142242442A11008844221")
900 CALL CHAR(124,"0054002A0
114810B4B3186327879330100AA0
054802882D0C28C1E5E0CA0C08")
910 CALL CHAR(128,"00787C3C1
E0F0F0F0F0F07070703030100 1E3
E3C78F0F0F0F0F0E0E0E0C0C08")
920 CALL CHAR(132,"010702030
60F1B3B6BFF030303060E0280E04
0C060F0D8DCD6FFC0C0C060704")
930 CALL CLEAR
940 CALL CHAR(116,RPT$("F",1
6),35,RPT$("F",16))
950 CALL CHAR(40,"80C0E0F0F8
FCFEFF",41,"0103070F1F3F7FFF
",42,"FF99E799E799E799")
960 FOR I=1 TO 40
970 !---------------------
980 !   BILDSCHIRMAUFBAU
990 !---------------------
1000 RANDOMIZE :: CALL HCHAR
((23*RND)+1,(31*RND)+1,(3*RN
D)+112):: NEXT I
1010 CALL HCHAR(24,1,116,32)
:: CALL HCHAR(23,7,116,18)::
 CALL HCHAR(22,7,116,18):: C
ALL HCHAR(21,7,116,18):: CAL
L HCHAR(20,7,116,18)
1020 CALL HCHAR(19,23,118)::
 CALL HCHAR(19,8,119):: CALL
 HCHAR(19,9,116,14):: CALL H
CHAR(18,11,119):: CALL HCHAR
(18,12,116,8)
1030 CALL HCHAR(18,20,118)::
 CALL COLOR(11,15,1):: CALL
HCHAR(23,1,35,6):: CALL HCHA
R(22,1,35,6):: CALL HCHAR(21
,1,35,6)
1040 CALL HCHAR(20,1,33,6)::
 CALL HCHAR(20,2,34):: CALL
HCHAR(20,4,34):: CALL HCHAR(
20,6,34):: CALL HCHAR(20,25,
33,8):: CALL HCHAR(20,26,34)
1050 CALL HCHAR(20,28,34)::
CALL HCHAR(20,30,34):: CALL
```

```
HCHAR(20,32,34):: CALL COLOR
(1,5,1)
1060 CALL HCHAR(21,25,35,8):
: CALL HCHAR(22,25,35,8):: C
ALL HCHAR(23,25,35,8)
1070 CALL HCHAR(17,17,40)::
CALL HCHAR(17,14,41):: CALL
HCHAR(16,15,41):: CALL HCHAR
(16,16,40):: CALL HCHAR(17,1
5,42,2):: CALL COLOR(2,9,1)
1080 EN=1000
1090 CALL HCHAR(17,12,104)::
 CALL HCHAR(17,19,105):: CAL
L HCHAR(16,12,108):: CALL HC
HAR(16,19,109):: CALL HCHAR(
15,13,106)
1100 CALL MAGNIFY(3)
1110 CALL HCHAR(15,18,107)::
 CALL HCHAR(14,14,110,4)
1120 DISPLAY AT(21,6)SIZE(15
):"ENERGIE:";EN
1130 DISPLAY AT(22,6)SIZE(15
):"SCORE  :";SC
1140 CALL SPRITE(#1,120,8,85
,114)
1150 CALL DELSPRITE(#2):: FO
R I=1 TO 90 :: NEXT I
1160 !---------------------
1170 ! ZUFAELLIGE BESTIMMUNG
1180 ! DER GESCHWINDIGKEIT
1190 ! UND DES AUSSEHENS
1200 ! DES FEINDLICHEN
1210 !   RAUMSCHIFFES
1220 !---------------------
1230 RANDOMIZE
1240 XXX=-(INT(30*RND)+6)
1250 SD=INT(4*RND+1)
1260 CALL CHAR(132,M$(SD))
1270 CALL SPRITE(#2,132,INT(
13*RND)+3,(80*RND)+1,1,2,XXX
)
1280 HIT=0
1290 CALL CHAR(33,A$,34,B$)
1300 GOSUB 1440
1310 CALL PATTERN(#1,120)::
CALL COLOR(#1,8)
1320 CALL KEY(0,C2,D2):: IF
C2=32 THEN CALL COLOR(10,11,
1):: SCH=1 :: EN=EN-100 ELSE
 CALL COLOR(10,1,1):: SCH=0
1330 CALL JOYST(1,C1,D1):: C
ALL MOTION(#1,-D1*3.5,C1*3.5
)
1340 CALL COINC(#2,115,114,2
0,HITS):: IF HITS THEN IF SC
H=0 THEN CALL DELSPRITE(#2):
: CALL SOUND(1000,-6,0):: CA
LL HCHAR(16,15,96,2)
1350 IF HITS=0 THEN 1400
1360 IF HITS AND SCH=1 THEN
1390
1370 IF HITS THEN CALL HCHAR
(17,14,98,4):: CALL COLOR(9,
9,1):: GOTO 1600
1380 GOTO 1400
1390 CALL PATTERN(#2,96):: C
ALL SOUND(500,-6,0):: FOR I=
1 TO 50 :: NEXT I :: GOTO 11
20
1400 CALL CHAR(34,A$,33,B$)
1410 GOSUB 1440
1420 DISPLAY AT(21,6)SIZE(15
):"ENERGIE:";EN :: IF EN<=0
THEN 1590
1430 GOTO 1290
1440 CALL JOYST(1,C1,D1)
1450 CALL POSITION(#2,A1,B1)
:: IF A1>125 THEN 1150
1460 CALL MOTION(#1,-D1*3.5,
C1*3.5)
1470 CALL KEY(1,C5,D5):: IF
C5=18 THEN 1520
1480 RETURN
1490 !--------------------
1500 !     SCHUSS !!!
1510 !--------------------
1520 CALL COLOR(2,9,1):: EN=
EN-10 :: CALL COINC(#1,#2,10
,HIT):: IF HIT THEN CALL PAT
TERN(#2,96):: CALL SOUND(300
,-5,0):: SC=SC+10 :: CALL MO
TION(#1,0,0)
1530 IF HIT=0 THEN CALL PATT
ERN(#1,100):: CALL COLOR(#1,
9)
1540 DISPLAY AT(22,6)SIZE(15
):"SCORE:";SC
1550 IF SC>190 THEN 1740
1560 IF HIT THEN 1150
1570 CALL JOYST(1,C1,D1):: C
ALL MOTION(#1,-D1*3.5,C1*3.5
)
1580 CALL COLOR(2,15,1):: RE
TURN
1590 HITS=-1 :: CALL SOUND(-
1000,-6,0):: CALL HCHAR(16,1
5,96,2):: CALL DELSPRITE(ALL
):: GOTO 1370
1600 CALL CLEAR
1610 !--------------------
1620 !     NIEDERLAGE  !!
1630 !--------------------
1640 DISPLAY AT(7,7):"DIE FE
INDLICHEN TRUPPEN  HA-BEN DI
CH BESIEGT": :"DEINE BASIS W
URDE ZERSTOERT": : : :"WILLS
T DU EINE ANDERE BASIS AUF A
TLANTIS VERTEIDIGEN ?"
1650 !--------------------
1660 !   NOCHMAL ABFRAGE
1670 !--------------------
```

```
1680 CALL DELSPRITE(ALL):: D
ISPLAY AT(17,1):"J=JA/N=NEIN
:" :: ACCEPT AT(17,14)SIZE(1
)VALIDATE("JNjn")BEEP:A$
1690 IF A$="J" OR A$="j" THE
N GOTO 310
1700 CALL CLEAR :: END
1710 !---------------------
1720 !        SIEG !!!
1730 !---------------------
1740 DISPLAY AT(7,7):"DU HAS
T UEBER DIE FEIND-.....LICHE
N ARMEEN GESIEGT!": :"BRAVO!
!!!!!!!!!!!!!!!!": : : :"MOEC
HTEST DU DIESEM PLANETENNOCH
 EINMAL HELFEN?"
1750 CALL CHAR(124,"10101010
10001010"):: CALL COLOR(12,1
6,1):: GOTO 1680
1760 CALL COLOR(2,16,1):: CA
LL CLEAR :: DISPLAY AT(1,8):
"ANLEITUNG:":TAB(8);"-------
---": :"DEINE AUFGABE IST ES
 EINE...BASIS AUF DEM WASSER
PLANETEN"
1770 DISPLAY AT(6,1):"ATLANT
IS ZU VERTEIDIGEN": :"LENKE
DEN CURSOR MIT DEINEM..JOYST
ICK AUF DAS ANGREIFENDERAUMS
CHIFF UND DRUECKE FEUER"
1780 DISPLAY AT(12,1):"SOLLT
EST DU DAS RAUMSCHIFF..NICHT
 TREFFEN SO KANNST DU..IM RI
CHTIGEN MOMENT FUER....EINIG
E SEKUNDEN EINEN"
1790 DISPLAY AT(16,1):"SCHUT
ZSCHIRM AKTIVIEREN,DER DICH
ALLERDINGS 100 ENERGIE-EINHE
ITEN KOSTET.EIN SCHUSS KOSTE
T NUR 10 EINHEITEN"
1800 DISPLAY AT(20,1):"ZWANZ
IG SCHIFFE GREIFEN.....INSGE
SAMMT AN,DANN HAST DU..SIE B
ESIEGT":"BITTE TASTE DRUECKE
N"
1810 CALL KEY(0,K,S):: IF S=
0 THEN 1810
1820 CALL CLEAR :: GOTO 700
```



```
100 ! **********************
110 ! *                    *
120 ! *  KOMPLEXE ZAHLEN   *
130 ! *                    *
140 ! *    Copyright by    *
150 ! *                    *
160 ! *  Gerhard Huebner   *
170 ! *                    *
180 ! *                    *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole  *
210 ! *     Ext. Basic     *
220 ! *                    *
260 ! *  Speicherbelegung  *
270 ! *     5337 Bytes     *
280 ! *                    *
290 ! **********************
300 ON WARNING NEXT
310 CALL CLEAR :: CALL COLOR
(14,7,1):: CALL HCHAR(5,2,13
8,30):: CALL VCHAR(6,32,139)
:: CALL VCHAR(6,2,139,1):: C
ALL HCHAR(7,2,137,30)
320 CALL HCHAR(20,2,138,30):
: CALL VCHAR(21,2,137,3):: C
ALL VCHAR(21,32,139,3):: CAL
L HCHAR(24,2,137,30)
330 ZF=10 :: ZE=1
340 CALL CHAR(130,"COFOFCFFF
FFCFOC",131,"0",132,"0018307
E7E3C1800",91,"00002442FF422
4",80,"0102040810204OFF")
350 CALL CHAR(137,"FFFF",138
,"000000000000FFFF",139,"COC
OCOCOCOCOCOC")
360 DISPLAY AT(1,2):"1.einga
be von z=re+j(im)":" 2.einga
be von z=r P"
370 DISPLAY AT(3,2):"3.einga
be der anzeige":" 4.eingabe
aus s:"
380 DISPLAY AT(ZF,2):"1.z=z
(pol [ kar)" :: DISPLAY AT(Z
F+1,2):"2.z=z konjungiet kom
plex"
390 DISPLAY AT(ZF+2,2):"3.z=
1/z1" :: DISPLAY AT(ZF+3,2):
"4.z=wurzel(z1)" :: DISPLAY
AT(ZF+4,2):"5.z in den speic
her s"
400 DISPLAY AT(ZF+5,2):"6.z=
z1+z2" :: DISPLAY AT(ZF+6,2)
:"7.z=z1-z2" :: DISPLAY AT(Z
F+7,2):"8.z=z1*z2" :: DISPLA
Y AT(ZF+8,2):"9.z=z1/z2"
410 !
420 !**F U N K T I O. W A H
L
430 CALL HCHAR(ZF,3,130)
440 CALL KEY(0,K,S):: IF S=0
 THEN GOTO 440
```

```
450 CALL HCHAR(ZF,3,131)
460 IF K=69 THEN ZF=ZF-1 ::
IF ZF=9 THEN ZF=10
470 IF K=88 THEN ZF=ZF+1 ::
IF ZF=19 THEN ZF=18
480 IF K=13 THEN CALL SOUND(
10,2000,0):: CALL HCHAR(ZF,3
,132):: CALL GCHAR(ZF,4,FU):
: GOTO 510
490 GOTO 430
500 !
510 IF FU=49 OR FU=50 OR FU=
51 OR FU=52 OR FU=53 THEN EI
N=1 ELSE EIN=2
520 GOSUB 1120
530 IF FU=49 THEN GOSUB 690
540 IF FU=50 THEN GOSUB 750
550 IF FU=51 THEN GOSUB 820
560 IF FU=52 THEN GOSUB 390
570 IF FU=53 THEN GOSUB 640
580 IF FU=54 THEN GOSUB 970
590 IF FU=55 THEN A(I-1)=-A(
I-1):: B(I-1)=-B(I-1):: GOSU
B 970
600 IF FU=56 THEN GOSUB 1040
610 IF FU=57 THEN C(I-1)=1/C
(I-1):: D(I-1)=-D(I-1):: GOS
UB 1040
620 GOTO 430
630 !
640 !**            ** SPEICHER
650 AS=A(I-1):: BS=B(I-1)::
CS=C(I-1):: DS=D(I-1)
660 DISPLAY AT(6,1):"s:" ::
DISPLAY AT(6,3)SIZE(12):AS :
: DISPLAY AT(6,15):"+J" :: D
ISPLAY AT(6,17)SIZE(12):BS
670 RETURN
680 !
690 ! **        **POLAR<>KARTE
S
700 DISPLAY AT(23,1):""
710 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
720 DISPLAY AT(22,1):"Z=" ::
 DISPLAY AT(22,3):C(I-1):: D
ISPLAY AT(22,16):"P" :: DISP
LAY AT(22,17):D(I-1)
730 RETURN
740 !
750 !**          ** VORZEICHEN
760 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
770 CALL VORZ(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
780 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
790 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
800 RETURN
810 !
820 !**          **   KEHRWERT
830 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
840 CALL REZI(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
850 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
860 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
870 RETURN
880 !
890 !**          **    WURZEL
900 DISPLAY AT(23,1):"" :: D
ISPLAY AT(22,1):""
910 !
920 CALL WURZ(A(I-1),B(I-1),
C(I-1),D(I-1),E1,F1,G1,H1)
930 A(I-1)=E1 :: B(I-1)=F1 :
: C(I-1)=G1 :: D(I-1)=H1
940 DISPLAY AT(21,1):"Z=" ::
 DISPLAY AT(21,3):A(I-1):: D
ISPLAY AT(21,15):"+J" :: DIS
PLAY AT(21,17):B(I-1)
950 RETURN
960 !
970 !**           **  ADDIERE
980 A(1)=A(1)+A(2):: B(1)=B(
1)+B(2)
990 CALL KARPOL(A(1),B(1),C(
1),D(1))
1000 DISPLAY AT(23,1):"" ::
DISPLAY AT(22,1):""
1010 DISPLAY AT(21,1):"Z=" :
: DISPLAY AT(21,3):A(1):: DI
SPLAY AT(21,15):"+J" :: DISP
LAY AT(21,17):B(1)
1020 RETURN
1030 !
1040 !**          **MULTIPLIZI
R
1050 C(1)=C(1)*C(2):: D(1)=D
(1)+D(2)
1060 CALL POLKAR(A(1),B(1),C
(1),D(1))
1070 DISPLAY AT(23,1):"" ::
DISPLAY AT(22,1):""
1080 DISPLAY AT(21,1):"Z=" :
: DISPLAY AT(21,3):A(1):: DI
SPLAY AT(21,15):"+J" :: DISP
LAY AT(21,17):B(1)
1090 RETURN
1100 !
1110 !****E I N G A B E ****
**
```

```
1120 FOR I=1 TO EIN
1130 CALL HCHAR(ZE,3,130)
1140 CALL KEY(0,K,S):: IF S=
0 THEN GOTO 1140
1150 CALL HCHAR(ZE,3,131)
1160 IF K=69 THEN ZE=ZE-1 ::
 IF ZE=0 THEN ZE=1
1170 IF K=88 THEN ZE=ZE+1 ::
 IF ZE=5 THEN ZE=4
1180 IF K=13 THEN CALL SOUND
(10,2000,0):: CALL GCHAR(ZE,
4,FE):: GOTO 1230
1190 GOTO 1130
1200 NEXT I
1210 RETURN
1220 !
1230 IF FE=49 THEN GOTO 1270
1240 IF FE=50 THEN GOTO 1300
1250 IF FE=51 THEN A(I+1)=A(
I):: B(I+1)=B(I):: C(I+1)=C(
I):: D(I+1)=D(I):: GOTO 1200
1260 IF FE=52 THEN A(I)=AS :
: B(I)=BS :: C(I)=CS :: D(I)
=DS :: GOTO 1200
1270 DISPLAY AT(23,1):"Z=...
.........+J..........." ::
DISPLAY AT(22,1):""
1280 DISPLAY AT(21,1):"Z=" :
: ACCEPT AT(21,3)SIZE(12)BEE
P:A(I):: ACCEPT AT(21,17)BEE
P:B(I)
1290 CALL KARPOL(A(I),B(I),C
(I),D(I)):: GOTO 1200
1300 DISPLAY AT(23,1):"Z=...
..........P..........." ::
DISPLAY AT(22,1):""
1310 DISPLAY AT(21,1):"Z=" :
: ACCEPT AT(21,3)SIZE(13)BEE
P:C(I):: ACCEPT AT(21,17)BEE
P:D(I)
1320 CALL POLKAR(A(I),B(I),C
(I),D(I)):: GOTO 1200
1330 !
1340 CALL POKA(A(I),B(I),C(I
),D(I),E(I),F(I),G(I),H(I))
1350 DISPLAY AT(19,1)SIZE(3)
:"Z=" :: DISPLAY AT(19,3):G(
I):: DISPLAY AT(19,16)SIZE(1
):"P" :: DISPLAY AT(19,17):H
(I)
1360 DISPLAY AT(20,1)SIZE(2)
:"Z=" :: DISPLAY AT(20,3):E(
I):: DISPLAY AT(20,15)SIZE(2
):"+J" :: DISPLAY AT(20,17):
F(I)
1370 !*********SUB POKA*****
**
1380 SUB POKA(A,B,C,D,E,F,G,
H)
1390 IF C=0 AND D=0 THEN CAL
L KARPOL(A,B,G,H):: E=A :: F
=B
1400 IF A=0 AND B=0 THEN CAL
L POLKAR(E,F,C,D):: G=C :: H
=D
1410 SUBEND
1420 !*********SUB POLKAR***
*
1430 SUB POLKAR(A,B,C,D)
1440 A=C*(COS(D*PI/180))
1450 B=C*(SIN(D*PI/180))
1460 SUBEND
1470 !***********SUB KARPOL*
*
1480 SUB KARPOL(A,B,C,D)
1490 C=SQR(A^2+B^2)
1500 IF A=0 AND B=0 THEN D=0
 :: GOTO 1560
1510 IF A=0 AND SGN(B)=1 THE
N D=90 :: GOTO 1560
1520 IF A=0 AND SGN(B)=-1 TH
EN D=-90 :: GOTO 1560
1530 IF B=0 AND SGN(A)=-1 TH
EN D=180 :: GOTO 1560
1540 D=ATN(B/A)*180/PI
1550 IF SGN(A)=-1 AND SGN(B)
=-1 OR(SGN(A)=-1 AND SGN(B)=
1)THEN D=D+180
1560 SUBEND
1570 !
1580 !***********SUB REZI***
1590 SUB REZI(A,B,C,D,E,F,G,
H)
1600 G=1/C :: H=-D
1610 CALL POLKAR(E,F,G,H)
1620 SUBEND
1630 !
1640 !************SUB WURZ**
1650 SUB WURZ(A,B,C,D,E,F,G,
H)
1660 G=SQR(ABS(C))
1670 H=D/2
1680 CALL POLKAR(E,F,G,H)
1690 SUBEND
1700 !
1710 !**********SUB VORZ***
1720 SUB VORZ(A,B,C,D,E,F,G,
H)
1730 B=-B
1740 CALL KARPOL(A,B,G,H)
1750 E=A :: F=B
1760 SUBEND
```

RESET

Eine Basic-Pokeliste für das Assemblerprogramm „RESET". Benötigt werden das Ex-Basic Modul und die 32 K RAM Expansion. Die Routine wird im Direktmoduls mit CALL LINK („RESET") gestartet. Dieses Programm sucht und findet ein „gelöschtes" Basicprogramm in der 32 K RAM Expansion, setzt die notwendigen Zeige und startet dieses Programm automatisch. Damit ein ordnungsgemäßer Ablauf erfolgt, müssen einige Voraussetzungen erfüllt sein. Dies sind:

1. Das Basicprogramm muß über mindestens eine Zeile mit einer Zeilennummer größer als 255 verfügen. Dies dürfte aber in der Regel kein Problem sein.
2. Die zuersg gespeicherte (eingegebene) Zeile sollte eine Kommentarzeile sein, da sie eventuell verändert wird. Dies ist nicht immer identisch mit der kleinsten Zeilennummer. Zum Umstellen gibt es folgende Möglichkeiten:

a Für Diskettenbetrieb:
1) Eingabe einer Kommentarzeile mit kleinster Zeilennummer un und mindestens 20 (in Worten zwanzig) Zeichen.
2) Speichern des Programmes mit SAVE DSK1.PRG,MERGE
3) Eingabe von NEW
4) Eingabe von MERGE DSK1.PRG
5) Speichern des Programmes mit SAVE DSK1.PRG

b Alle anderen müssen einen umständlicheren Weg einschlagen:
1) Siehe a) 1)
2) Editieren des gesamten Programmes, d.h., nacheinander über die Cursortasten jede Zeile anwählen und dort mindestens ein Zeichen überschreiben. Die ENTER Taste zu drücken reicht nicht aus.

Es versteht sich ja fast von selbst, daß diese Kommentarzeile nicht mehr geändert werden darf. Beim Eingeben neuer Programme zuerst die Kommentarzeile und dann das restliche Programm eingeben.

Hier nun noch eine kurze Routine, die ein versehentliches Ändern erschwert. Im Direktmodus eingeben:
CALL PEEK (−32718, A,B)::CALL LOAD (A*256+B−65539,0,0)

Zur großen Überraschung hat die erste Zeile nun die Nummer 0 und ist daher gegen Ändern geschützt. Durch Eingabe von „RES" erhält diese Zeile wieder einen normalen Wert.

Peter Vorwerk

Disassembler

Der Disassembler ist ein nützliches Werkzeug für Minimem-Benutzer, das aber auch für Extended Basic und/oder Memory Expansion geeignet ist. Man kann damit Maschinensprache-Routinen disassemblen und somit auch von Programmen, die mit dem Line-by-Line Assembler geschrieben sind, ein Listing erzeugen. Dabei werden die Symbole der symbolischen Adressen und die Namen der Minimem Utility Routinen eingesetzt.

Natürlich kann man auch beliebige Betriebssystem-Routinen des ROM übersetzen.

Außerdem können mit diesem Programm RAM/ROM-Bereiche in hexadezimaler Form angezeigt werden, ebenso Bereiche im VDP-RAM.

Als Zugabe gibt es die Möglichkeit, das RAM/ROM nach einem vierstelligen Hexwert zu durchsuchen.

Zum schnelleren Ablauf steht zum Umrechnen der Dezimalzahlen wahlweise eine Routine in Maschinensprache zur Verfügung, die jedoch 80 Bytes im Minimem belegt.

Als Hardcopy wird ein 80-stelliger Drucker über eine RS232-Schnittstelle benutzt.

Nach dem Starten erscheint ein Auswahl-Menü mit 6 Optionen:

1. Disassembler allgemein
2. Disassembler mit Symbolen und Utility-Namen
3. Auffinden eines Hexstrings
4. Dumpen aus RAM/ROM
5. Dumpen aus VDP-RAM
6. Programmende

Nachdem eine Option gewählt worden ist, erfolgen die Eingaben für Drucker und Maschinensprache-Routine (ja/nein). Danach werden die Anfangs- und Endadresse des gewünschten Bereiches eingegeben. Nach Erreichen der Endadresse wird gefragt, ob mit der gleichen Option weitergemacht oder das Programm beendet werden soll.

Bis auf Option 5 laufen alle Optionen auch unter Extended Basic, allrdings ist Option 2 wegen der fehlenden Tabellen nicht sinnvoll. Ebenso kann die Maschinensprachroutine natürlich nicht geladen werden, es sei denn, amn hat eine Memexp und ändert die Adressen. Das Format der Ausgabe ist aus den Mustern ersichtlich. Die Bildschirmausgabe ist ähnlich (zwei Zeilen pro Assembler-Befehl, 3 statt 8 2-Byte Blöcke beim Dump).

Erdbeer-Paul

Die Hauptrolle des Programms spielt Paul, der Zwerg. Seine Aufgabe besteht im Pflücken von 12 Erdbeeren. Diese kann er aufnehmen, indem er unter die Erdbeere geht und Taste Q gedrückt wird. Hat er eine Erdbeere aufgenommen, so wird dieses am unteren rechten Bildrand angezeigt. Die Erdbeere muß er dann zu dem Topf bringen. Steht er am am Topf, muß noch einmal die Taste S bedient werden, damit Paul die Erdbeere ablegt. Aber natürlich gibt es auch Gefahren: Eine Maus in der unteren Etage, Spinnen in der oberen und eine Biene. Stößt Paul mit einem dieser Tiere zusammen, ist das Spiel beendet (neuer Start: Drücken einer Taste). Paul. hab aber auch eine Chance, denn er besitzt 5 Schwerter, mit diesen kann er alle Tiere außer der Biene vertreiben. Pro vertriebenem Tier gibt es 15 Punkte, pro abgelegter Erdbeere 30 Punkte und 6 Zusatzschwerte. Sinkt die Zahl der Schwerte auf 0, dann ist das Spiel beendet. Sind alle Erdbeeren geerntet, beginnt eine neue Runde. Das Benutzen der Schwerter erfolgt über Taste , :links und Taste. :rechts.

Gunther Krämling

```
      BF EQU >7F60
      BG EQU >7F66
      BY EQU >7F94
      GN EQU >7FB4
      SC EQU >7FC6
      NM EQU >7FD6

7F66  C28B       BG MOV  R11,R10
7F68  04C0          CLR  R0
7F6A  0201 0001     LI   R1,1
7F6E  0205 7F60     LI   R5,BF
7F72  0206 0400     LI   R6,>0400
7F76  DD46          MOVB R6,*R5+
7F78  06A0 7FB4     BL   @GN
7F7C  C204          MOV  R4,R8
7F7E  0581          INC  R1
7F80  06A0 7FB4     BL   @GN
7F84  0288 0001     CI   R8,1
7F88  1605          JNE  BY
7F8A  06A0 7FC6     BL   @SC
7F8E  0A44          SLA  R4,4
7F90  06A0 7FC6     BL   @SC
7F94  0A84       BY SLA  R4,8
7F96  06A0 7FC6     BL   @SC
7F9A  0AC4          SLA  R4,12
7F9C  06A0 7FC6     BL   @SC
7FA0  0581          INC  R1
7FA2  0288 0001     CI   R8,1
7FA6  131D          JEQ  >7FE2
7FA8  0281 0004     CI   R1,4
7FAC  131A          JEQ  >7FE2
7FAE  06A0 7FB4     BL   @GN
7FB2  10F0          JMP  BY
7FB4  0420 6044  GN BLWP @NUMREF
7FB8  0420 601C     BLWP @XMLLNK
7FBC  1200          JLE  >7FBE
7FBE  C120 834A     MOV  @>834A,R4
7FC2  C104          MOV  R4,R7
7FC4  045B          B    *R11
7FC6  0884       SC SRA  R4,8
7FC8  D100          MOVB R0,R4
7FCA  0844          SRA  R4,4
7FCC  0284 000A     CI   R4,10
7FD0  1102          JLT  NM
7FD2  0224 0007     AI   R4,7
7FD6  0224 0030  NM AI   R4,48
7FDA  0A84          SLA  R4,8
7FDC  DD44          MOVB R4,*R5+
7FDE  C107          MOV  R7,R4
7FE0  045B          B    *R11
7FE2  0202 7F60     LI   R2,BF
7FE6  0420 6048     BLWP @STRASG
7FEA  045A          B    *R10
```



```
10 REM*******************
11 REM*                 *
12 REM*   DISASSEMBLER  *
13 REM*   Copyright by  *
14 REM*                 *
15 REM* Fritz-H. Mueller *
16 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*  Minimem-Modul  *
20 REM*optional: Xbasic *
21 REM*  32-K erweiterung*
22 REM*  Drucker        *
23 REM* Speicherbelegung *
24 REM*     944Ø Bytes   *
25 REM*******************
26 REM
1ØØ CALL CLEAR
11Ø CALL SCREEN(15)
12Ø PRINT : : : : : : : :"
     DISASSEMBLER": : :
13Ø PRINT "   (C) F.H.MUELLE
R 1984 ": : : : : :
14Ø REM  AM BOERNEKEN 1
15Ø REM  3181 RUEHEN
16Ø REM  TEL.Ø5367/886
17Ø DIM OP$(78),UNAM$(28),UA
D(28),SY$(4Ø),SYA(4Ø)
18Ø DEF TX(X)=INT(X-((X>=1Ø)
*7)+48)
19Ø DEF RST(Y)=Y-INT(Y)
2ØØ DEF UH(X)=INT(X/16)
21Ø DEF LH(X)=X-(UH(X)*16)
22Ø DEF UB2=INT(B(2)/64)
23Ø DEF LB2=B(2)-(UB2*64)
24Ø DEF TS=INT(LB2/16)
25Ø DEF SAD=B(1)*256+B(2)
26Ø DATA >," ","  ",",",(,),
R,*,-,+,,@
27Ø READ ZX$,L1$,L2$,ZK$,KA$
,KZ$,R$,ST$,M$,PL$,LL$,KLA$
28Ø DATA COC,CZC,XOR,XOP,LDC
R,STCR,MPY,DIV
29Ø DATA LIMI,IDLE,RSET,RTWP
,CKON,CKOF,LREX
3ØØ DATA JMP,JLT,JLE,JEQ,JHE
,JGT,JNE,JNC,JOC,JNO,JL,JH,J
OP,SBO,SBZ,TB
31Ø DATA LI,,AI,,ANDI,,ORI,,
CI,,STWP,,STST,,LWPI
32Ø DATA BLWP,NEG,DEC,SETO,B
,INV,DECT,ABS,X,INC,BL,,CLR,
INCT,SWPB,
33Ø DATA SRA,SRL,SLA,SRC
34Ø DATA SZC,SZCB,S,SB,C,CB,
A,AB,MOV,MOVB,SOC,SOCB
35Ø FOR I=1 TO 78
36Ø READ OP$(I)
37Ø NEXT I
38Ø DATA 1,3,Ø,8,11,16,Ø,Ø,"
  "
```

```
390 DATA 1,8,0,20,25,17,"  "
400 GOTO 1500
410 REM ***** COVERT BYTE TO
 HEX *****
420 ON AN3 GOTO 430,450
430 CALL LINK("TOX$",2,B(1),
B(2),BX$)
440 RETURN
450 FOR I=1 TO 2
460 A$=CHR$(TX(B(I)/16))
470 H$=CHR$(TX(RST(B(I)/16)*
16))
480 AH$(I)=A$&H$
490 NEXT I
500 BX$=AH$(1)&AH$(2)
510 RETURN
520 REM ***** CONVERT ADDRES
S TO HEX *****
530 ON AN3 GOTO 540,560
540 CALL LINK("TOX$",1,ADI,B
X$)
550 RETURN
560 B(1)=ADI/256
570 B(2)=256*(RST(B(1)))
580 GOTO 450
590 REM
600 REM ***** CONVERT TO DEC
IMAL *****
610 REM
620 ADO=0
630 FOR I=1 TO 4
640 AAS(I)=ASC(SEG$(AXI$,I,1
))
650 IF AAS(I)>57 THEN 680
660 AD(I)=VAL(CHR$(AAS(I)))
670 GOTO 690
680 AD(I)=AAS(I)-55
690 ADO=ADO+(AD(I)*16^(4-I))
700 NEXT I
710 RETURN
720 REM ***** INCR ADDRESS A
ND PEEK 2 BYTES *****
730 ADR=ADR+2
740 ADRS=ADRS+2
750 IF ADRS>=ADRE THEN 3300
760 IF ADR<32768 THEN 780
770 ADR=ADR-65536
780 IF AC=1 THEN 860
790 ON AN3 GOTO 800,820
800 ADI=ADR
810 GOTO 830
820 ADI=ADRS
830 GOSUB 520
840 ADR$=BX$
850 PRINT #P:ADR$;LD$;
860 IF OPT>4 THEN 920
870 CALL PEEK(ADR,B(1),B(2))
880 IF OPT<>3 THEN 930
890 IF SAD<>SU THEN 730
900 GOSUB 4060
910 GOTO 730
920 CALL PEEKV(ADR,B(1),B(2)
)
930 GOSUB 410
940 PRINT #P:BX$;L1$;
950 IF SYT=0 THEN 1010
960 IF RR=1 THEN 1000
970 CAD=ADR
980 GOSUB 3940
990 RETURN
1000 GOSUB 3910
1010 IF OPT>3 THEN 1040
1020 RETURN
1030 REM  ***** DUMP FORMAT
SUBROUTINE *****
1040 AC=1
1050 FOR I=1 TO 2
1060 IF B(I)<32 THEN 1080
1070 IF B(I)<95 THEN 1090
1080 B(I)=46
1090 BP$(I)=CHR$(B(I))
1100 NEXT I
1110 BC$(BL)=BP$(1)&BP$(2)
1120 IF BL=BM THEN 1150
1130 BL=BL+1
1140 GOTO 730
1150 IF P=0 THEN 1170
1160 PRINT #P:L2$;
1170 FOR I=1 TO BM
1180 PRINT #P:BC$(I);
1190 NEXT I
1200 PRINT #P
1210 BL=1
1220 AC=0
1230 GOTO 730
1240 REM ***** REGISTER SUBR
TN *****
1250 RD$=R$&STR$(RD)
1260 ON RIX GOTO 1270,1290,1
310,1450
1270 OP1$=RD$&KO$
1280 RETURN
1290 OP1$=ST$&RD$&KO$
1300 RETURN
1310 RR=1
1320 AC=1
1330 GOSUB 730
1340 IF OI=47 THEN 1350 ELSE
 1380
1350 UF=0
1360 GOSUB 3820
1370 IF UF=1 THEN 1410
1380 IF SF=1 THEN 1400
1390 HX$=KLA$&ZX$
1400 IF RD<>0 THEN 1430
1410 OP1$=HX$&BX$&KO$
1420 RETURN
1430 OP1$=HX$&BX$&KA$&RD$&KZ
$&KO$
1440 RETURN
1450 OP1$=ST$&RD$&PL$&KO$
1460 RETURN
1470 REM   *
```

```
             *
1480 REM  ***** BEGINN HAUPT
PROGRAMM *****
1490 REM   *
             *
1500 RESTORE 380
1510 READ BL,BM,AC,TL,TC,TP,
OPT,SYT,LD$
1520 PRINT "WAEHLE AUS :": :
1530 PRINT "DISASSEMBLER = 1
":
1540 PRINT "DISASS M/SYM = 2
":
1550 PRINT "MEMORY SCAN  = 3
":
1560 PRINT "DUMP.........= 4
":
1570 PRINT "DUMP VDP.....= 5
":
1580 PRINT "PROGRAMMENDE = 6
": : :
1590 INPUT OPT
1600 PRINT
1610 IF OPT<>6 THEN 1630
1620 END
1630 IF OPT=3 THEN 1690
1640 INPUT "DRUCKEN N=0,J=1
":P
1650 IF P=0 THEN 1690
1660 OPEN #P:"RS232",OUTPUT
1670 RESTORE 390
1680 READ BL,BM,AC,TC,TP,TL,
LD$
1690 INPUT "ASM J=1,N=2 ":AN
3
1700 PRINT
1710 IF AN3=2 THEN 1730
1720 GOSUB 4130
1730 CALL CLEAR
1740 PRINT
1750 INPUT "BEGINN,ENDE (HEX
) ":ADR$,ADRE$
1760 PRINT
1770 IF LEN(ADR$)<>4 THEN 17
90
1780 IF LEN(ADRE$)=4 THEN 18
20
1790 PRINT "* 4 CHAR HEXWERT
 EINGEBEN "
1800 CALL SOUND(100,200,3)
1810 GOTO 1750
1820 ON OPT GOTO 1990,1960,1
850,1990,1990
1840 REM **** FIND WITHIN RA
M/ROM ****
1850 INPUT "SUCHBEGRIFF (HEX
) ":AXI$
1860 IF LEN(AXI$)=4 THEN 190
0
1870 PRINT "* SUCHBEGRIFF MU
SS VIERSTELLIG SEIN"
1880 CALL SOUND(100,200,3)
1890 GOTO 1850
1900 GOSUB 620
1910 SU=ADO
1920 AC=1
1940 GOTO 1990
1950 REM *** SETUP INITIAL A
DDRESSES ***
1960 IF SYT=1 THEN 1990
1970 IF UT=1 THEN 1990
1980 GOSUB 3430
1990 AXI$=ADRE$
2000 GOSUB 620
2010 ADRE=ADO
2020 AXI$=ADR$
2030 GOSUB 620
2040 ADR=ADO
2050 ADRS=ADO
2060 RR=0
2070 GOSUB 760
2080 GOTO 2170
2090 REM **** FORMAT ASSEMBL
ER OUTPUT ****
2100 PRINT #P:TAB(TL);LB$;
2110 PRINT #P:TAB(TC);OPC$;T
AB(TP);OP1$;OP2$
2120 RR=0
2130 LB$=LL$
2140 AC=0
2150 GOSUB 730
2160 REM **** FIND OPCODE **
**
2170 OP1$=LL$
2180 OP2$=LL$
2190 HX$=ZX$
2200 KO$=ZK$
2210 B1=INT(B(1)/4)+1
2220 IF B(1)>15 THEN 2250
2230 ON B1 GOTO 2240,2580,26
80,2290
2240 ON B(1)+1 GOTO 2290,229
0,2330,2480
2250 IF B(1)<32 THEN 2740
2260 IF B(1)<64 THEN 2940
2270 GOTO 3080
2280 REM ***** INVALID OPCOD
E IS DATA *****
2290 OPC$=LE1$
2300 PRINT #P:TAB(TC);"DATA"
;L1$;HX$;BX$
2310 GOTO 2120
2320 REM  ***** FORMAT VIII
INSTRUCTIONS *****
2330 OI=UH(B(2))+32
2340 OPC$=OP$(OI)
2350 IF OPC$=LL$ THEN 2290
2360 IF OPC$="LWPI" THEN 238
0
2370 OP1$=R$&STR$(LH(B(2)))&
KO$
2380 AC=1
```

```
2390 RR=1
2400 GOSUB 730
2410 IF B(1)>0 THEN 2440
2420 OP2$=STR$(B(2))
2430 GOTO 2100
2440 IF SF=0 THEN 2460
2450 HX$=LL$
2460 OP2$=HX$&BX$
2470 GOTO 2100
2480 REM ***** FORMAT VII IN
STRUCTIONS *****
2490 IF B(2)>0 THEN 2520
2500 OPC$=OP$(9)
2510 GOTO 2380
2520 F=INT(B(2)/32)
2530 IF B(2)-(F*32)<>0 THEN
2290
2540 OI=F+8
2550 OPC$=OP$(OI)
2560 GOTO 2100
2570 REM ***** FORMAT VI INS
TRUCTIONS *****
2580 H=B(1)+43
2590 OI=H+(UB2*4)
2600 OPC$=OP$(OI)
2610 IF OPC$=LL$ THEN 2290
2620 RD=LH(B(2))
2630 KO$=LL$
2640 RIX=TS+1
2650 GOSUB 1250
2660 GOTO 2100
2670 REM ***** FORMAT V INST
RUCTIONS *****
2680 OI=B(1)+55
2690 OPC$=OP$(OI)
2700 OP1$=R$&STR$(LH(B(2)))&
KO$
2710 OP2$=STR$(UH(B(2)))
2720 GOTO 2100
2730 REM ***** FORMAT II INS
TRUCTIONS *****
2740 OI=B(1)
2750 OPC$=OP$(OI)
2760 IF SEG$(OPC$,1,1)="J" T
HEN 2790
2770 OP1$=STR$(B(2))
2780 GOTO 2100
2790 IF B(2)>127 THEN 2820
2800 ADI=ADR+(B(2)*2)+2
2810 GOTO 2830
2820 ADI=ADR-((255-B(2))*2)
2830 IF SYT=0 THEN 2900
2840 RR=1
2850 CAD=ADI
2860 GOSUB 3930
2870 IF SF=0 THEN 2900
2880 HX$=LL$
2890 GOTO 2910
2900 GOSUB 530
2910 OP1$=HX$&BX$
2920 GOTO 2100
2930 REM ***** FORMAT III/IV
/IX INSTRUCTIONS *****
2940 OI=INT((B(1)-28)/4)
2950 OPC$=OP$(OI)
2960 RD1=(B(1)-(28+(OI*4)))*
4
2970 RD=RD1+UB2
2980 IF LEN(OPC$)=4 THEN 304
0
2990 OP2$=R$&STR$(RD)
3000 RD=LH(B(2))
3010 RIX=TS+1
3020 GOSUB 1250
3030 GOTO 2100
3040 OP2$=STR$(RD)
3050 GOTO 3000
3060 REM
3070 REM  ***** FORMAT I INS
TRUCTIONS *****
3080 OI=UH(B(1))+63
3090 OPC$=OP$(OI)
3100 B1S=B(1)
3110 B2S=B(2)
3120 RIX=TS+1
3130 RD=LH(B(2))
3140 KO$=LL$
3150 GOSUB 1250
3160 OP1S$=OP1$
3170 B(1)=B1S
3180 B(2)=B2S
3190 TD=INT(LH(B(1))/4)
3200 B1L=(LH(B(1))-(TD*4))*4
3210 RIX=TD+1
3220 RD=B1L+UB2
3230 GOSUB 1250
3240 OP2$=OP1$
3250 OP1$=OP1S$&ZK$
3260 GOTO 2100
3270 REM  *
   *
3280 REM ***** ENDE PROGM **
***
3290 REM  *
   *
3300 PRINT #P
3310 PRINT
3320 INPUT "WEITER ? J/N ":A
$
3330 IF A$="N" THEN 3400
3340 IF P=0 THEN 3370
3350 RESTORE 390
3360 GOTO 3380
3370 RESTORE 380
3380 READ BL,BM,AC
3390 GOTO 1740
3400 IF P=0 THEN 1500
3410 CLOSE #P
3420 GOTO 1500
3430 REM **** EINLESEN UTILI
TY TABLE *****
3440 UADR=28438
```

```
3450 PRINT :"UTLTAB... ";
3460 FOR I=1 TO 28
3470 CALL PEEK(UADR,N(1),N(2
),N(3),N(4),N(5),N(6),B(1),B
(2))
3480 FOR J=1 TO 6
3490 NA$(J)=CHR$(N(J))
3500 NEXT J
3510 UNAM$(I)=NA$(1)&NA$(2)&
NA$(3)&NA$(4)&NA$(5)&NA$(6)
3520 UAD(I)=SAD
3530 UADR=UADR+8
3540 NEXT I
3550 UT=1
3560 PRINT "EINGELESEN": :
3570 IF OPT=2 THEN 3600
3580 RETURN
3590 REM ***** EINLESEN SYMB
OL TABLE *****
3600 SYCA=29072
3610 SYTA=31960
3620 SYT=1
3630 CALL PEEK(SYCA,B(1),B(2
))
3640 IF SAD>1 THEN 3670
3650 SYT=0
3660 RETURN
3670 SYC=SAD
3680 FOR V=1 TO SYC
3690 CALL PEEK(SYTA,N(1),N(2
),B(1),B(2))
3700 FOR J=1 TO 2
3710 S$(J)=CHR$(N(J))
3720 NEXT J
3730 SY$(V)=S$(1)&S$(2)
3740 SYA(V)=SAD
3750 GOSUB 420
3760 PRINT #P:TAB(6);SY$(V);
L1$;"EQU";L1$;ZX$;BX$
3770 SYTA=SYTA+4
3780 NEXT V
3790 PRINT #P
3800 RETURN
3810 REM **** SCAN UTLTAB **
**
3820 IF UT=0 THEN 3860
3830 FOR I=1 TO 28
3840 IF SAD=UAD(I)THEN 3870
3850 NEXT I
3860 RETURN
3870 BX$=UNAM$(I)
3880 HX$=KLA$
3890 UF=1
3900 RETURN
3910 REM **** SCAN SYMTAB **
**
3920 CAD=SAD
3930 SF=0
3940 FOR I=1 TO SYC
3950 IF CAD=SYA(I)THEN 3980
3960 NEXT I
3970 RETURN
3980 SF=1
3990 IF RR=1 THEN 4020
4000 LB$=SY$(I)
4010 RETURN
4020 BX$=SY$(I)
4030 RR=0
4040 GOTO 3880
4050 REM ***** PRINT GEF.SUC
HBEGIFF ****
4060 ADI=ADR
4070 GOSUB 420
4080 SU$=BX$
4090 GOSUB 530
4100 PRINT #P:BX$;L1$;ZX$;SU
$
4110 GOTO 730
4120 REM ***** TEST U LADEN
ASM PROG *****
4130 DATA 84,79,88,36,32,32,
127,102
4140 PAD=32760
4150 RESTORE 4130
4160 FOR I=0 TO 3
4170 READ A
4180 CALL PEEK(PAD+I,NM)
4190 IF NM<>A THEN 4220
4200 NEXT I
4210 RETURN
4220 PRINT "** TOX$ PGM NICH
T GELADEN. **"
4230 PRINT "LADEN BELEGT >7F
60 BIS >7FF0 "
4240 PRINT "UND REF/DEF TABL
E >7FF8 ": :
4250 INPUT "OK? J/N ":AN$
4260 IF AN$="N" THEN 4270 EL
SE 4290
4270 AN3=2
4280 RETURN
4290 DATA 194,139,4,192,2,1,
0,1,2,5,127,96,2,6,4,0,221,7
0,6,160
4300 DATA 127,180,194,4,5,12
9,6,160,127,180,2,136,0,1,22
,5,6,160,127,198
4310 DATA 10,68,6,160,127,19
8,10,132,6,160,127,198,10,19
6,6,160,127,198,5,129
4320 DATA 2,136,0,1,19,29,2,
129,0,4,19,26,6,160,127,180,
16,240,4,32
4330 DATA 96,68,4,32,96,28,1
8,0,193,32,131,74,193,196,4,
91,8,132,209,0
4340 DATA 8,68,2,132,0,10,17
,2,2,36,0,7,2,36,0,48,10,132
,221,68
4350 DATA 193,7,4,91,2,2,127
,96,4,32,96,72,4,90
4360 LP=7
```

```
4370 RESTORE 4130
4380 FOR I=0 TO LP
4390 READ BY
4400 CALL LOAD(PAD+I,BY)
4410 NEXT I
4420 IF LP=133 THEN 4470
4430 PAD=32614
4440 LP=133
4450 RESTORE 4290
4460 GOTO 4380
4470 PRINT
4472 CALL LOAD(28702,127,248
)
4480 RETURN
```



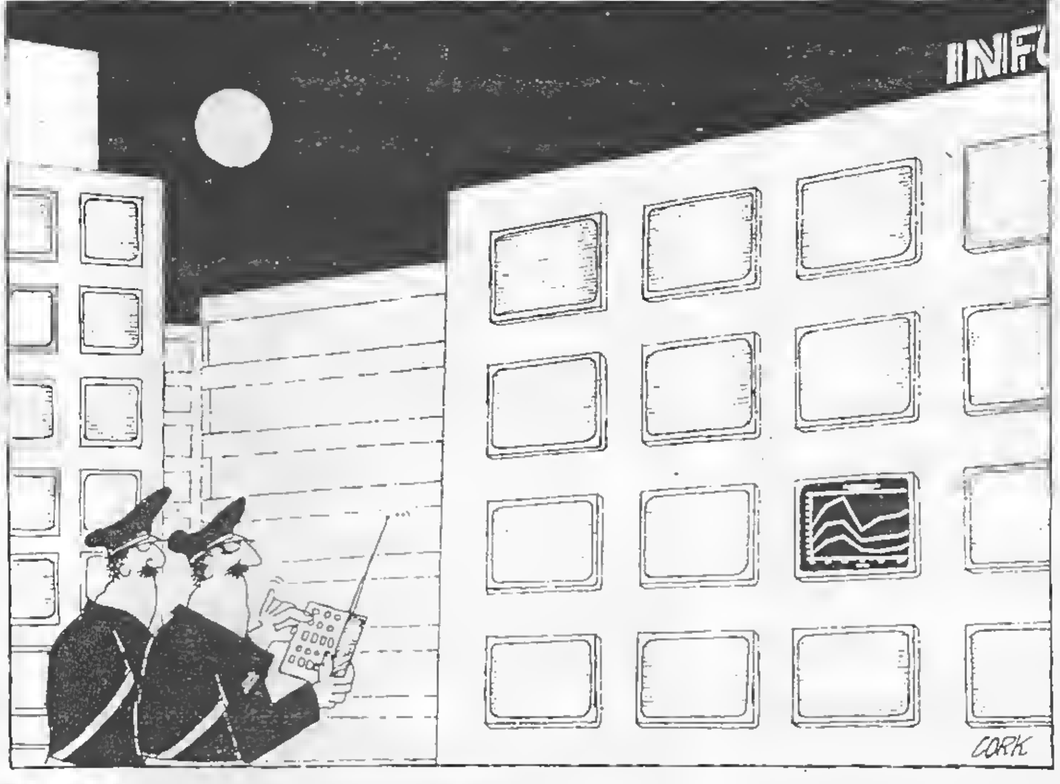

```
10 ! **********************
11 ! *                    *
12 ! *       RESET        *
13 ! *                    *
14 ! *   Copyright by     *
15 ! *                    *
16 ! *   Peter Vorwerk    *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *  32-K Erweiterung  *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     1599 Bytes     *
28 ! *                    *
29 ! **********************
30 !
100 RESTORE 350 :: FOR I=1 T
O 10 :: READ CH(I):: NEXT I
110 RESTORE :: FOR I=1 TO 10
 :: SUM=0 :: FOR J=1 TO 16 :
: READ A :: SUM=SUM+A :: NEX
T J
120 IF SUM<>CH(I)THEN PRINT
"* FEHLER IN ZEILE";I*10+130
 :: CALL SOUND(100,110,0)::
STOP
130 NEXT I
140 PRINT " ALLES O.K.":" JE
TZT WIRD GEPOKED"
150 RESTORE
160 ADR=15398
170 CALL INIT
180 READ A :: IF A<>-1 THEN
CALL LOAD(ADR,A):: ADR=ADR+1
 :: GOTO 180
190 CALL LOAD(8194,60,186,63
,248)
200 FOR I=16376 TO 16383 ::
READ A :: CALL LOAD(I,A):: N
EXT I
210 DATA 0,99,2,224,60,0,2,1
,255,216,6,65,6,65,216,49
220 DATA 60,36,216,49,60,34,
216,49,60,32,152,32,60,36,60
,38
230 DATA 22,244,152,32,60,34
,60,38,22,240,152,32,60,32,6
0,38
240 DATA 22,236,200,1,60,36,
2,33,0,2,216,49,60,34,216,49
250 DATA 60,35,2,33,0,2,216,
49,60,32,216,49,60,33,152,32
260 DATA 60,32,60,36,17,13,1
52,32,60,32,60,34,21,242,17,
4
270 DATA 152,32,60,33,60,35,
20,237,200,32,60,32,60,34,16
,233
280 DATA 6,96,60,35,200,32,6
0,36,131,49,200,32,60,34,128
,51
290 DATA 216,32,60,39,131,35
,216,32,60,38,131,124,2,224,
131,224
300 DATA 4,96,0,112,255,0,25
5,0,255,0,255,0,255,0,255,0
310 DATA -1
320 REM NAMENSTABELLE:
330 DATA 82,69,83,69,84,32,6
0,40,0,0,0,0,0,0,0,0
340 REM PRUEFSUMMEN
350 DATA 1266,1190,1278,1216
,1031,872,1296,1210,1695,174
2
```

```
10 REM*******************
11 REM*                 *
12 REM*  ERDBEER-PAUL   *
13 REM*  Copyright  by  *
14 REM*Gunther  Kraemling*
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*    6194 Bytes    *
25 REM*******************
26 REM
200 CALL CLEAR
210 CALL CHAR(33,"FFFFFFFFFF
FFFFFF")
220 CALL CHAR(34,"FFFFFF")
230 CALL CHAR(40,"FFFFFFFFFF
FFFFFF")
240 CALL CHAR(58,"7E5A7E18FF
182442")
250 CALL CHAR(59,"00007E5AFF
99A5A5")
260 CALL CHAR(64,"060C9C4830
080808")
270 CALL CHAR(72,"0C1E1E0EE4
BFFF0E")
280 CALL CHAR(80,"3C243C243C
243C24")
290 CALL CHAR(88,"00000C0201
2D7EFE")
300 CALL CHAR(89,"187E181818
3C7EFF")
310 CALL CHAR(96,"3F3F070F1F
3F7FFF")
320 CALL CHAR(97,"FCFCE0F0F8
FCFEFF")
330 CALL CHAR(98,"FFFFFFFFFF
7F3F1F")
340 CALL CHAR(102,"FFFFFFFFF
FFEFCF8")
350 CALL CHAR(106,"0000004CF
C40")
360 CALL CHAR(107,"000000023
F02")
370 CALL CHAR(120,"0E1F3F3F1
E0E")
380 CALL CHAR(128,"0000387E7
E7E3C18")
390 CALL SCREEN(7)
400 CALL COLOR(1,3,1)
410 CALL COLOR(2,14,1)
420 CALL COLOR(3,2,1)
430 CALL COLOR(4,2,1)
440 CALL COLOR(5,3,1)
450 CALL COLOR(6,12,1)
460 CALL COLOR(7,11,1)
470 CALL COLOR(8,15,1)
480 CALL COLOR(9,6,1)
490 CALL COLOR(10,6,1)
500 CALL COLOR(11,6,1)
510 CALL COLOR(12,9,1)
520 CALL COLOR(13,9,16)
530 CALL HCHAR(3,1,40,32)
540 CALL HCHAR(20,1,33,32)
550 CALL HCHAR(8,1,40,16)
560 CALL VCHAR(9,16,40,9)
570 CALL HCHAR(10,17,34,3)
580 CALL HCHAR(10,21,34,7)
590 CALL HCHAR(10,29,34,4)
600 CALL HCHAR(15,1,34,4)
610 CALL HCHAR(15,7,34,7)
620 CALL HCHAR(15,15,34)
630 CALL HCHAR(15,17,34)
640 CALL HCHAR(15,19,34,11)
650 CALL HCHAR(15,31,34,2)
660 CALL HCHAR(18,5,34,2)
670 CALL VCHAR(10,20,80,4)
680 CALL VCHAR(10,28,80,4)
690 CALL VCHAR(15,14,80,4)
700 CALL VCHAR(15,18,80,4)
710 CALL VCHAR(15,30,80,4)
720 CALL VCHAR(16,5,96)
730 CALL VCHAR(16,6,97)
740 CALL VCHAR(17,5,98)
750 CALL VCHAR(17,6,102)
760 HP=0
770 B=24
780 C=32
790 X=19
800 Y=3
810 A=7
820 P=0
830 D=4
840 E=24
850 F=5
860 R=24
861 U=0
870 FOR I=19 TO 31 STEP 3
880 CALL HCHAR(8,I,64)
890 CALL HCHAR(8,I-1,120)
900 NEXT I
910 CALL HCHAR(13,19,64)
920 CALL HCHAR(13,18,120)
930 CALL HCHAR(13,23,64)
940 CALL HCHAR(13,22,120)
950 CALL HCHAR(13,26,64)
960 CALL HCHAR(13,25,120)
970 CALL HCHAR(13,31,64)
980 CALL HCHAR(13,30,120)
990 FOR I=21 TO 27 STEP 3
1000 CALL HCHAR(18,I,64)
1010 CALL HCHAR(18,I-1,120)
1020 NEXT I
1030 CALL HCHAR(14,A,59)
1040 CALL HCHAR(14,B,59)
1050 CALL HCHAR(9,R,59)
1060 CALL HCHAR(19,C,88)
1070 CALL HCHAR(D,E,72)
1080 GOSUB 2870
1090 GOSUB 2910
```

```
1100 GOSUB 2950
1110 CALL HCHAR(X,Y,58)
1120 CALL KEY(0,K,S)
1130 IF S=0 THEN 1570
1140 IF K<>68 THEN 1220
1150 IF Y=32 THEN 1590
1160 CALL GCHAR(X,Y+1,M)
1170 IF M=0 THEN 1200
1180 IF M=40 THEN 1590
1190 CALL HCHAR(X,Y,32)
1200 Y=Y+1
1210 GOTO 1570
1220 IF K<>83 THEN 1310
1230 IF Y=1 THEN 1590
1240 IF (Y=7)*(X=14)THEN 143
0
1250 CALL GCHAR(X,Y-1,M)
1260 IF M=0 THEN 1260
1270 IF M=40 THEN 1590
1280 CALL HCHAR(X,Y,32)
1290 Y=Y-1
1300 GOTO 1570
1310 IF K<>69 THEN 1370
1320 CALL GCHAR(X-1,Y,M)
1330 IF M<>80 THEN 1590
1340 CALL HCHAR(X,Y,32)
1350 X=X-5
1360 GOTO 1570
1370 IF K<>88 THEN 1590
1380 CALL GCHAR(X+1,Y,M)
1390 IF M<>80 THEN 1590
1400 CALL HCHAR(X,Y,32)
1410 X=X+5
1420 GOTO 1570
1430 IF N=0 THEN 1570
1440 FOR I=1 TO 3
1450 CALL SOUND(150,-I,6)
1460 NEXT I
1470 P=P+30
1480 GOSUB 2870
1490 F=F+6
1500 GOSUB 2910
1510 N=0
1511 U=U+1
1512 IF U=12 THEN 861
1520 CALL HCHAR(21,27,32)
1530 CALL HCHAR(14,6,120)
1540 CALL HCHAR(14,6,32)
1550 CALL HCHAR(15,5,120)
1560 CALL HCHAR(15,5,32)
1570 CALL HCHAR(X,Y,58)
1580 CALL SOUND(20,600,6)
1590 IF K<>81 THEN 1660
1600 CALL GCHAR(X-1,Y,M)
1610 IF M<>120 THEN 1660
1620 CALL HCHAR(X-1,Y,32)
1630 CALL HCHAR(21,27,128)
1640 N=1
1650 CALL SOUND(50,2000,5)
1660 IF K<>46 THEN 1800
1670 IF Y=32 THEN 2120
1680 CALL GCHAR(X,Y+1,M)
1690 IF M=40 THEN 2120
1700 IF M=32 THEN 1760
1710 CALL HCHAR(X,Y+1,106)
1720 F=F-1
1730 CALL HCHAR(X,Y+1,32)
1740 GOTO 1930
1750 F=F-1
1760 CALL HCHAR(X,Y+1,106)
1770 F=F-1
1780 CALL HCHAR(X,Y+1,32)
1790 GOTO 2100
1800 IF K<>44 THEN 2120
1810 IF Y=1 THEN 2120
1820 CALL GCHAR(X,Y-1,M)
1830 IF M=40 THEN 2120
1840 IF M=32 THEN 1890
1850 CALL HCHAR(X,Y-1,107)
1860 F=F-1
1870 CALL HCHAR(X,Y-1,32)
1880 GOTO 1930
1890 CALL HCHAR(X,Y-1,107)
1900 F=F-1
1910 CALL HCHAR(X,Y-1,32)
1920 GOTO 2100
1930 IF X<>19 THEN 1970
1940 C=32
1950 CALL HCHAR(19,C,88)
1960 GOTO 2060
1970 IF X<>9 THEN 2010
1980 R=32
1990 CALL HCHAR(9,R,59)
2000 GOTO 2060
2010 IF Y<16 THEN 2050
2020 B=32
2030 CALL HCHAR(14,B,59)
2040 GOTO 2060
2050 A=1
2060 CALL HCHAR(14,A,59)
2070 CALL SOUND(150,110,6)
2080 P=P+15
2090 GOSUB 2870
2100 GOSUB 2910
2110 IF F=0 THEN 2650
2120 IF X<>19 THEN 2220
2130 IF Y<C THEN 2170
2140 C=C+1
2150 CALL HCHAR(19,C-1,32)
2160 GOTO 2190
2170 C=C-1
2180 CALL HCHAR(19,C+1,32)
2190 CALL HCHAR(19,C,88)
2200 IF C=Y THEN 2650
2210 GOTO 1120
2220 IF X<>9 THEN 2500
2230 IF Y>R THEN 2270
2240 R=R-1
2250 CALL HCHAR(9,R+1,32)
2260 GOTO 2290
2270 R=R+1
```

```
2280 CALL HCHAR(9,R-1,32)
2290 CALL HCHAR(9,R,59)
2300 IF R=Y THEN 2650
2310 IF Y>E THEN 2380
2320 CALL GCHAR(D+1,E-1,M)
2330 IF (M=32)+(M=58)THEN 23
40 ELSE 2440
2340 D=D+1
2350 E=E-1
2360 CALL HCHAR(D-1,E+1,32)
2370 GOTO 2470
2380 CALL GCHAR(D+1,E+1,M)
2390 IF (M=32)+(M=58)THEN 24
00 ELSE 2440
2400 D=D+1
2410 E=E+1
2420 CALL HCHAR(D-1,E-1,32)
2430 GOTO 2470
2440 CALL HCHAR(D,E,32)
2450 D=4
2460 C=24
2470 CALL HCHAR(D,E,72)
2480 IF (E=Y)*(D=X)THEN 2650
2490 GOTO 1120
2500 IF Y<16 THEN 2600
2510 IF Y<B THEN 2550
2520 B=B+1
2530 CALL HCHAR(14,B-1,32)
2540 GOTO 2570
2550 B=B-1
2560 CALL HCHAR(14,B+1,32)
2570 CALL HCHAR(14,B,59)
2580 IF B=Y THEN 2650
2590 GOTO 1120
2600 A=A+1
2610 CALL HCHAR(14,A-1,32)
2620 CALL HCHAR(14,A,59)
2630 IF A=Y THEN 2650
2640 GOTO 1120
2650 CALL HCHAR(X,Y,89)
2660 FOR I=70 TO 100
2670 CALL SOUND(10,210-I,5)
2680 NEXT I
2690 FOR I=1 TO 500
2700 CALL KEY(0,K,S)
2710 IF S<>0 THEN 2750
2720 NEXT I
2730 CALL CLEAR
2740 END
2750 CALL HCHAR(X,Y,32)
2760 CALL HCHAR(14,A,32)
2770 CALL HCHAR(14,B,32)
2780 CALL HCHAR(9,R,32)
2790 CALL HCHAR(19,C,32)
2800 CALL HCHAR(D,E,32)
2810 CALL HCHAR(21,27,32)
2820 CALL HCHAR(23,13,32,5)
2830 CALL HCHAR(24,13,32,5)
2840 IF P<=HP THEN 770
2850 HP=P
2860 GOTO 770
2870 U$="score>"&STR$(P)&"
   "
2880 V=23
2890 W=6
2900 GOTO 2980
2910 U$="swords>"&STR$(F)&"
   "
2920 V=24
2930 W=5
2940 GOTO 2980
2950 U$="highscore>"&STR$(HF
)&""
2960 V=22
2970 W=2
2980 FOR I=1 TO LEN(U$)
2990 Z=ASC(SEG$(U$,I,1))
3000 CALL HCHAR(V,W,Z)
3010 W=W+1
3020 NEXT I
3030 RETURN
```

BÖRSE



DIE HAND WERKER KOMMEN

Soft-Plot

Soft-Plot dient zur schnellen Erstellung einfarbiger Graphiken und mehrere Graphik-Funktionen erleichtern dabei die Arbeit.
1. Radier-Funktionen: löscht jeweils ein 4*4 Dots großes Quadrat
2. Vergrößern: 4-fache Vergrößerung des Zeichens.
3. Kreis-Funktion: Plottet Kreise mit beliebigen Radien.
4. Linien-Funktion: Plottet eine Linie zwischen Anfangs- und Endpunkt. Gleichmäßige Punkteverteilung.
5. Char-Funktion: Kopieren eines Zeichens, um freie Zeichen zu sparen.
6. Lösch-Funktion: Löscht den Bildschirm und setzt alle Zeichen wieder auf „0".
7. Ausdruck: Programm Ende. Ausdruck des Hexcodes sowie der Position D. Zeichen.
Ist der Zeichenvorrat erschöpft, kann man in den gesetzten Zeichen nachplotten.

Das Programm erklärt sich im wesentlichen selbst – will man die Erklärung jedoch weglassen, sind folgende Zeilen zu löschen:
> 220
> 240
> 1550 bis Ende

> 170 – Einleitung.
> 300 – Bildschirm Aufbau
> 450 – Option: Vergrößern
> 510 – Option: Radieren
> 610 – Option: Linien ziehen.
> 630 – Option: Kreise zeichnen
> 730 – Option: Zeichen plazieren
> 760 – Unterprogramm Linie
> 900 – Unterprogramm Plotten
> 1170 – Steuerung des Zeichenstiftes.
> 1230 – Wahl der Optionen
> 1420 – Anzeige der Anzahl freier Zeichen
> 1490 – Ausdruck der Zeichen und Programmende.
> 1550 – Erklärung des Programms
> 1710 – Textdaten der Erklärung

3x3 Matrix

Habt Ihr schon mal neidisch auf andere Computersysteme geschaut, in deren Betriebssprache sogar Matrizenbefehle integriert sind?

Für einen einfachen Befehl in der Textverarbeitung, z.B. 100 MAT A=RxS, müßt Ihr in TI- oder Ext.-Basic schon ein ganzes Programm schreiben, andere Befehle wie MAT READ, MAT INPUT usw. sind ebenfalls nur sehr umständlich in die Sprache des TI99 zu übersetzen. Mit diesem Programm wollen wir Euch zeigen, wie Ihr Euch MAT-Befehle für Euren TI schreiben könnt. Es kann 3 MAT-Statements in Verbindung mit einer 3x3 Matrix realisieren.
1.) MAT A=BxC, 2. MAT A=B+C 3.) MAT A=INV (B).

In die Felder der Matrizen dürft Ihr allerdings nur bis zu 3-stellige Zahlen (incl. Nachkommatrennzeichen) eingeben. Dabei müßt Ihr darauf achten, daß das Ergebnis höchstens vier Stellen besitzt, sonst liefert Euch der Computer nur Sternchen.

Ein Menü führt Euch durch das Programm.

Habt Ihr etwas Ahnung von Matrixoperationen, so seht Euch dieses Listing genau an. Wenn Ihr das Prinzip verstanden habt, dürfte es Euch eigentlich nicht schwer fallen, das Programm beliebig zu erweitern. Neidische Blicke auf andere Systeme gehören damit der Vergangenheit an.

Chr. Villwock

```
100 ! ***********************
110 ! *                     *
120 ! *     3X3 Matrix      *
130 ! *    Copyright by     *
140 ! * Christian Villwock  *
150 ! *                     *
160 ! *   Bearbeitet vom    *
170 ! *     Team des        *
180 ! *  Aktuell Verlages   *
190 ! *     Muenchen        *
200 ! *                     *
210 ! * Benoetigte Geraete  *
220 ! *  TI99/4A  Konsole   *
230 ! *     Ext. Basic      *
240 ! *                     *
270 ! *  Speicherbelegung   *
280 ! *     6756 Bytes      *
290 ! ***********************
295 !
300 CALL CLEAR :: CALL SCREE
N(2):: ON WARNING NEXT :: ON
 BREAK NEXT
310 FOR I=1 TO 13 :: CALL CO
LOR(I,16,1):: NEXT I
320 CALL CHAR(120,"3C4299A1A
199423C")
330 CALL COLOR(14,9,9)
340 R=3
350 M$="x 1984"
360 GOSUB 660
370 R=5
380 M$="PROGRAMM VON"
390 GOSUB 660
400 R=7
410 M$="CHRISTIAN VILLWOCK"
420 GOSUB 660
430 R=9
440 M$="--------------------
-"
450 GOSUB 660
460 R=10
470 M$="3 * 3 M A T R I Z E
N"
480 GOSUB 660
490 R=11
500 M$="--------------------
-"
510 GOSUB 660
520 R=13
530 M$="FUER"
540 GOSUB 660
550 R=15
560 M$="TI REVUE LESER"
570 GOSUB 660
580 R=17
590 M$="UND "
600 GOSUB 660
610 R=19
620 M$="ALLE ANDEREN"
630 GOSUB 660
640 FOR I=1 TO 1000 :: NEXT
I
650 GOTO 720
660 C=16-LEN(M$)/2
670 FOR I=1 TO LEN(M$)
680 CALL HCHAR(R,C+I,136)
690 CALL HCHAR(R,C+I,ASC(SEG
$(M$,I,1)))
700 NEXT I
710 RETURN
720 CALL CLEAR :: CALL SCREE
N(4)
730 FOR I=1 TO 14 :: CALL CO
LOR(I,2,1):: NEXT I
740 ! UMKEHRMATRIX
750 DEF AC=(A22*A33)-(A32*A2
3)
760 DEF BC=(-1)*((A21*A33)-(
A31*A23))
770 DEF CB=(A21*A32)-(A31*A2
2)
780 DEF DC=(-1)*((A12*A33)-(
A32*A13))
790 DEF EC=(A11*A33)-(A31*A1
3)
800 DEF FC=(-1)*((A11*A32)-(
A31*A12))
810 DEF GC=(A12*A23)-(A22*A1
3)
820 DEF HC=(-1)*((A11*A23)-(
A21*A13))
830 DEF IC=(A11*A22)-(A21*A1
2)
840 DEF DET=(A11*A22*A33)+(A
12*A23*A31)+(A21*A32*A13)-(A
13*A22*A31)-(A11*A23*A32)-(A
33*A12*A21)
850 ! MULTIPLIKATION
860 DEF AA=(A11*B11)+(A12*B2
1)+(A13*B31)
870 DEF BB=(A11*B12)+(A12*B2
2)+(A13*B32)
880 DEF CC=(A11*B13)+(A12*B2
3)+(A13*B33)
890 DEF DD=(A21*B11)+(A22*B2
1)+(A23*B31)
900 DEF EE=(A21*B12)+(A22*B2
2)+(A23*B32)
910 DEF FF=(A21*B13)+(A22*B2
3)+(A23*B33)
920 DEF GG=(A31*B11)+(A32*B2
1)+(A33*B31)
930 DEF HH=(A31*B12)+(A32*B2
2)+(A33*B32)
940 DEF II=(A31*B13)+(A32*B2
3)+(A33*B33)
950 ! ADDITION
960 DEF AB=A11+B11
970 DEF BA=A12+B12
980 DEF CA=A13+B13
990 DEF DA=A21+B21
```

```
1000 DEF EA=A22+B22
1010 DEF FA=A23+B23
1020 DEF GA=A31+B31
1030 DEF HA=A32+B32
1040 DEF IA=A33+B33
1050 ! ZEICHEN FUER TABELLE
1060 CALL CHAR(100,"000000FF
FF",101,"030303030303FFFF",1
02,"00000000000FFFF",104,"0
303037F7F030303",103,RPT$("0
3",8))
1070 CALL CHAR(105,"18181818
1818FFFF",106,"181818FFFF181
818",107,RPT$("18",8))
1080 CALL COLOR(9,2,1,10,2,1
)
1090 ! MENUE
1100 CALL CLEAR :: CALL SCRE
EN(4)
1110 DISPLAY AT(5,8):"3*3 MA
TRIZEN"
1120 DISPLAY AT(7,6):"------
-----------"
1130 DISPLAY AT(9,7):"PROGRA
MMWAHL :"
1140 DISPLAY AT(11,7):"1 =>
MULTIPLIKATION"
1150 DISPLAY AT(13,7):"2 =>
ADDITION"
1160 DISPLAY AT(15,7):"3 =>
UMKEHRMATRIX"
1170 DISPLAY AT(17,7):"4 =>
ENDE"
1180 DISPLAY AT(20,7):"BITTE
 WAEHLEN SIE :"
1190 ACCEPT AT(22,7)VALIDATE
("1234")BEEP SIZE(1):WAHL
1200 IF WAHL=1 THEN 1240 ELS
E IF WAHL=2 THEN 1240 ELSE I
F WAHL=3 THEN 1240 ELSE IF W
AHL=4 THEN 2590
1210 GOTO 1190
1220 ! ABFRAGE M1
1230 XYZ=1
1240 GOSUB 2040
1250 CALL SCREEN(6)
1260 DISPLAY AT(3,7):"1g..1.
.k..2..k..3..k"
1270 ACCEPT AT(6,10)VALIDATE
(NUMERIC)BEEP SIZE(3):A11
1280 ACCEPT AT(6,16)VALIDATE
(NUMERIC)SIZE(3):A12
1290 ACCEPT AT(6,22)VALIDATE
(NUMERIC)SIZE(3):A13
1300 ACCEPT AT(9,10)VALIDATE
(NUMERIC)BEEP SIZE(3):A21
1310 ACCEPT AT(9,16)VALIDATE
(NUMERIC)SIZE(3):A22
1320 ACCEPT AT(9,22)VALIDATE
(NUMERIC)SIZE(3):A23
1330 ACCEPT AT(12,10)VALIDAT
E(NUMERIC)BEEP SIZE(3):A31
1340 ACCEPT AT(12,16)VALIDAT
E(NUMERIC)SIZE(3):A32
1350 ACCEPT AT(12,22)VALIDAT
E(NUMERIC)SIZE(3):A33
1360 XYZ=1
1370 GOSUB 2620
1380 IF WAHL=3 THEN 2180
1390 CALL CLEAR
1400 ! ABFRAGE M2
1410 XYZ=2
1420 GOSUB 2040
1430 DISPLAY AT(3,7):"2g..1.
.k..2..k..3..k"
1440 ACCEPT AT(6,10)VALIDATE
(NUMERIC)BEEP SIZE(3):B11
1450 ACCEPT AT(6,16)VALIDATE
(NUMERIC)SIZE(3):B12
1460 ACCEPT AT(6,22)VALIDATE
(NUMERIC)SIZE(3):B13
1470 ACCEPT AT(9,10)VALIDATE
(NUMERIC)BEEP SIZE(3):B21
1480 ACCEPT AT(9,16)VALIDATE
(NUMERIC)SIZE(3):B22
1490 ACCEPT AT(9,22)VALIDATE
(NUMERIC)SIZE(3):B23
1500 ACCEPT AT(12,10)VALIDAT
E(NUMERIC)BEEP SIZE(3):B31
1510 ACCEPT AT(12,16)VALIDAT
E(NUMERIC)SIZE(3):B32
1520 ACCEPT AT(12,22)VALIDAT
E(NUMERIC)SIZE(3):B33
1530 IF WAHL=1 OR WAHL=2 THE
N GOSUB 2620
1540 IF WAHL=1 THEN 1560 ELS
E IF WAHL=2 THEN 1740
1550 !  MULTIPLIKATION
1560 GOSUB 2040
1570 DISPLAY AT(3,5):"1*2g..
1..k..2..k..3..k"
1580 DISPLAY AT(15,8):"MULTI
PLIKATION !"
1590 DISPLAY AT(6,9):USING "
####":AA
1600 DISPLAY AT(6,15):USING
"####":BB
1610 DISPLAY AT(6,21):USING
"####":BB
1620 DISPLAY AT(9,9):USING "
####":DD
1630 DISPLAY AT(9,15):USING
"####":EE
1640 DISPLAY AT(9,21):USING
"####":FF
1650 DISPLAY AT(12,9):USING
"####":GG
1660 DISPLAY AT(12,15):USING
"####":HH
1670 DISPLAY AT(12,21):USING
"####":II
```

```
1680 GOSUB 2690
1690 GOSUB 1900
1700 CALL KEY(0,K,S):: IF S=
0 THEN 1700 ELSE 1080
1710 GOTO 1700
1720 ! ADDITION
1730 GOSUB 2040
1740 DISPLAY AT(6,9):USING "
####":AB
1750 DISPLAY AT(3,5):"1+2g..
1..k..2..k..3..k"
1760 DISPLAY AT(15,11):"ADDI
TION !"
1770 DISPLAY AT(6,15):USING
"####":BA
1780 DISPLAY AT(6,21):USING
"####":CA
1790 DISPLAY AT(9,9):USING "
####":DA
1800 DISPLAY AT(9,15):USING
"####":EA
1810 DISPLAY AT(9,21):USING
"####":FA
1820 DISPLAY AT(12,9):USING
"####":GA
1830 DISPLAY AT(12,15):USING
 "####":HA
1840 DISPLAY AT(12,21):USING
 "####":IA
1850 GOSUB 2690
1860 GOSUB 1900
1870 CALL KEY(0,K,S):: IF S=
0 THEN 1870 ELSE 1080
1880 GOTO 1870
1890 ! DARSTELLUNG M1 U. M2
1900 DISPLAY AT(17,2):A11 ::
 DISPLAY AT(17,6):A12
1910 DISPLAY AT(17,10):A13 :
: DISPLAY AT(19,2):A21
1920 DISPLAY AT(19,6):A22 ::
 DISPLAY AT(19,10):A23
1930 DISPLAY AT(21,2):A31 ::
 DISPLAY AT(21,6):A32
1940 DISPLAY AT(21,10):A33 :
: DISPLAY AT(17,17):B11
1950 DISPLAY AT(17,21):B12 :
: DISPLAY AT(17,25):B13
1960 DISPLAY AT(19,17):B21 :
: DISPLAY AT(19,21):B22
1970 DISPLAY AT(19,25):B23 :
: DISPLAY AT(21,17):B31
1980 DISPLAY AT(21,21):B32 :
: DISPLAY AT(21,25):B33
1990 IF WAHL=1 THEN CALL HCH
AR(19,17,42)
2000 IF WAHL=2 THEN CALL HCH
AR(19,17,43)
2010 DISPLAY AT(24,2):"<< TA
STE DRUECKEN >>¬"
2020 ! ZEICHNEN DER TABELLE
2030 RETURN
2040 CALL CLEAR
2050 DISPLAY AT(3,8):"g..1..
...2.....3"
2060 DISPLAY AT(4,7):"feffff
fiffffffiffffffi"
2070 DISPLAY AT(5,8):"g.....
k.....k.....k"
2080 DISPLAY AT(6,7):"1h....
.k.....k.....k"
2090 DISPLAY AT(7,8):"gddddd
jddddddjddddddj"
2100 DISPLAY AT(8,8):"g.....
k.....k.....k"
2110 DISPLAY AT(9,7):"2h....
.k.....k.....k"
2120 DISPLAY AT(10,8):"gdddd
djddddddjddddddj"
2130 DISPLAY AT(11,8):"g....
.k.....k.....k"
2140 DISPLAY AT(12,7):"3h...
..k.....k.....k"
2150 DISPLAY AT(13,8):"gdddd
djddddddjddddddj"
2160 RETURN
2170 ! UMKEHRMATRIX
2180 GOSUB 2040
2190 DISPLAY AT(15,7):"UMKEH
RMATRIX !"
2200 DISPLAY AT(17,6):"DIE D
ETERMINANTE LAUTET :"
2210 DISPLAY AT(19,10):DET
2220 IF DET=0 THEN 2560
2230 DISPLAY AT(3,7):"Tg..1.
.k..2..k..3..k"
2240 DISPLAY AT(6,9):USING "
####":AC
2250 DISPLAY AT(6,15):USING
"####":BC
2260 DISPLAY AT(6,21):USING
"####":CB
2270 DISPLAY AT(9,9):USING "
####":DC
2280 DISPLAY AT(9,15):USING
"####":EC
2290 DISPLAY AT(9,21):USING
"####":FC
2300 DISPLAY AT(12,9):USING
"####":GC
2310 DISPLAY AT(12,15):USING
 "####":HC
2320 DISPLAY AT(12,21):USING
 "####":IC
2330 GOSUB 2690
2340 DISPLAY AT(24,2):"TASTE
 DRUECKEN FUER UMKEHR."
2350 CALL KEY(0,K,S):: IF S=
0 THEN 2350 ELSE 2380
2360 GOTO 2350
2370 ! SPIEGELUNG
2380 GOSUB 2040
```

```
2390 DISPLAY AT(3,8):"g..1..
k..2..k..3..k"
2400 DISPLAY AT(6,9):USING "
####":AC
2410 DISPLAY AT(6,15):USING
"####":DC
2420 DISPLAY AT(6,21):USING
"####":GC
2430 DISPLAY AT(9,9):USING "
####":BC
2440 DISPLAY AT(9,15):USING
"####":EC
2450 DISPLAY AT(9,21):USING
"####":HC
2460 DISPLAY AT(12,9):USING
"####":CB
2470 DISPLAY AT(12,15):USING
 "####":FC
2480 DISPLAY AT(12,21):USING
 "####":IC
2490 GOSUB 2690
2500 DISPLAY AT(15,7):"UMKEH
RMATRIX !"
2510 DISPLAY AT(17,9):"DETER
MINANTE :"
2520 DISPLAY AT(19,10):DET
2530 DISPLAY AT(24,2):"<< TA
STE DRUECKEN >>"
2540 CALL KEY(0,K,S):: IF S=
0 THEN 2540 ELSE 1100
2550 GOTO 2540
2560 DISPLAY AT(21,2):"DIE D
ETERMINANTE IST UNGUEL-TIG,
DA 0=0."
2570 GOTO 2230
2580 ! ENDE
2590 CALL CLEAR
2600 DISPLAY AT(9,5):"BIS ZU
M NAECHSTEN MAL."
2610 END
2620 DISPLAY AT(24,2):"KORRI
GIEREN <J> <N>"
2630 CALL KEY(0,K,S):: IF S=
0 THEN 2630
2640 IF K=78 OR K=110 THEN 2
680
2650 IF K=74 OR K=106 THEN 2
660 ELSE 2670
2660 IF XYZ=1 THEN 1220 ELSE
 IF XYZ=2 THEN 1400
2670 GOTO 2630
2680 RETURN
2690 FOR I=6 TO 12 STEP 3 ::
 CALL VCHAR(I,10,104):: CALL
 VCHAR(I,16,107):: CALL VCHA
R(I,22,107):: CALL VCHAR(I,2
8,107):: NEXT I
2700 RETURN
```

```
100 !
110 ! **********************
120 ! *                    *
130 ! *    >SOFT-PLOT<     *
140 ! *                    *
150 ! *   Copyright by     *
160 ! *                    *
170 ! *   Kay  Sievert     *
180 ! *                    *
190 ! *                    *
200 ! * Benoetigte Geraete *
210 ! *  TI99/4A  Konsole  *
220 ! *     Ext. Basic     *
230 ! *    Joystick (1)    *
260 ! *                    *
270 ! *  Speicherbelegung  *
280 ! *    11264 Bytes     *
290 ! *                    *
      **********************
300 GOTO 370 :: CALL CLEAR :
: CALL SCREEN :: CALL KEY ::
 CALL JOYST :: CALL LOCATE :
: CALL MAGNIFY :: CALL SPRIT
E :: CALL SOUND :: CALL COLO
R :: CALL HCHAR :: CALL VCHA
R :: CALL GCHAR :: CALL DELS
PRITE
310 CALL CHARPAT :: CALL CHA
RSET :: CALL PATTERN
320 DIM Z$(10),CH$(100)
330 A :: B :: C :: D :: E ::
 F :: H :: I :: K :: L :: M
:: N :: O :: Q :: S :: U ::
V :: W :: X :: Y :: PR :: PR
S :: NR :: NC :: CN :: ROW :
: COL :: RA :: CA
340 CX :: CY :: CXM :: CYM :
: D1X :: D2X :: D1Y :: D2Y :
: XS :: YS :: XIS :: YIS ::
MA :: MB :: RAD :: ST :: PO
350 X$ :: Y$ :: C$
360 !@P-
370 CALL CLEAR :: CALL MAGNI
FY(2)
380 CALL SPRITE(#1,83,5,49,7
0,#2,79,5,49,86,#3,70,5,49,1
02,#4,84,5,49,116)
390 CALL SPRITE(#5,80,5,70,1
00,#6,76,5,70,116,#7,79,5,70
,132,#8,84,5,70,148)
400 X$="* (C)1985 by  KAY SI
EVERT *"
410 FOR I=1 TO 27 :: CALL HC
HAR(20,2+I,ASC(SEG$(X$,I,1))
):: CALL SOUND(-25,880,9)::
NEXT I
420 DISPLAY AT(24,4):"ERKLAE
RUNG ?(J/N):N" :: ACCEPT AT(
24,22)VALIDATE("JN")SIZE(-1)
:X$
430 DISPLAY AT(24,2):"[ALPHA
 LOCK]-TASTE LOESEN !" :: CA
```

```
LL SOUND(1000,110,10):: FOR
I=1 TO 500 :: NEXT I
440 IF X$="J" THEN A=1 :: GO
TO 450
450 CALL DELSPRITE(ALL):: CA
LL MAGNIFY(1):: CALL CLEAR
460 FOR I=0 TO 9 :: CALL CHA
RPAT(48+I,X$):: Z$(I)=X$ ::
NEXT I
470 C$="0000000100100011010O
0101011001111000100110101011
1100110111101111"
480 W,F=1 :: PR,PRS=4 :: NR=
44 :: PO=2.5
490 !***** BILDSCHIRM ******
500 CALL SCREEN(1):: CALL HC
HAR(1,1,31,768)
510 !@P+
520 CALL CHAR(32,"0000000000
0000FF",33,"0101010101010101
",34,"FF",35,"8080808080808080
80")
530 CALL CHAR(36,"02070E1C38
70608",37,"040E1F3E7CF8702",
38,"FF818181813181FF",39,"FF
C7BBBFBFBFBBC7")
540 CALL CHAR(40,"3C42818131
81423C",41,"AA55AA55AA55AA55
",42,"FOFOFOFO")
550 !@P-
560 CALL HCHAR(1,3,32,28)::
CALL HCHAR(23,3,34,28):: CAL
L VCHAR(2,2,33,21):: CALL VC
HAR(2,27,33,21)
570 CALL VCHAR(2,31,35,21)::
 FOR I=5 TO 20 STEP 3 :: CAL
L HCHAR(I,28,34,3):: NEXT I
580 CALL HCHAR(13,28,30)
590 CALL HCHAR(3,29,37):: CA
LL HCHAR(6,29,38):: CALL HCH
AR(9,29,40):: CALL HCHAR(12,
29,34):: CALL HCHAR(15,29,39
)
600 CALL COLOR(0,6,1):: FOR
I=1 TO 14 :: CALL COLOR(I,4,
1):: NEXT I
610 IF A=1 THEN 1760
620 FOR I=45 TO 143 :: CALL
CHAR(I,"0"):: NEXT I :: GOSU
B 1640
630 ROW,COL=89 :: CALL SPRIT
E(#1,36,6,ROW-7,COL,#2,36,6,
200,100)
640 GOTO 1370
650 !**** VERGROESSERN *****
660 CALL GCHAR(INT(ROW/8),IN
T(COL/8)+1,V):: IF V=31 THEN
 1370
670 CALL CHARPAT(V,X$)
680 CALL CHAR(41,X$)
690 CALL SPRITE(#3,41,15,157
,221):: CALL MAGNIFY(2)
700 CALL KEY(1,Q,S):: IF Q<>
18 OR S=-1 THEN 700 ELSE CAL
L MAGNIFY(1):: CALL SPRITE(#
1,38,6,ROW-7,COL):: GOTO 137
0
710 !****  RADIEREN  ******
720 U=ROW/8 :: V=COL/8
730 IF U-INT(U)>.5 THEN X=1
ELSE X=2
740 IF V-INT(V)>.5 THEN Y=1
ELSE Y=2
750 CALL GCHAR(U,V,Q):: IF Q
=31 THEN 1370 ELSE CALL CHAR
PAT(Q,X$)
760 IF X=1 AND Y=1 THEN X$="
0"&SEG$(X$,2,1)&"0"&SEG$(X$,
4,1)&"0"&SEG$(X$,6,1)&"0"&SE
G$(X$,8,9):: GOTO 800
770 IF X=1 AND Y=2 THEN X$=S
EG$(X$,1,1)&"0"&SEG$(X$,3,1)
&"0"&SEG$(X$,5,1)&"0"&SEG$(X
$,7,1)&"0"&SEG$(X$,9,8):: GO
TO 800
780 IF X=2 AND Y=1 THEN X$=S
EG$(X$,1,8)&"0"&SEG$(X$,10,1
)&"0"&SEG$(X$,12,1)&"0"&SEG$
(X$,14,1)&"0"&SEG$(X$,16,1):
: GOTO 800
790 X$=SEG$(X$,1,9)&"0"&SEG$
(X$,11,1)&"0"&SEG$(X$,13,1)&
"0"&SEG$(X$,15,1)&"0"
800 CXM,CYM=0 :: CALL CHAR(Q
,X$):: GOTO 1370
810 !***** LINIE  ******
820 IF W=1 THEN A=ROW :: B=C
OL :: W=2 :: GOTO 1370 ELSE
IF W=2 THEN W=1 :: CALL LOCA
TE(#2,ROW-7,COL):: C=ROW ::
D=COL :: GOTO 970
830 !***** KREIS  *******
840 IF W=1 THEN A=ROW :: B=C
OL :: W=2 :: GOTO 1370 ELSE
IF W=2 THEN W=1 :: CALL LOCA
TE(#2,ROW-7,COL):: C=ROW ::
D=COL :: GOTO 850
850 CALL LOCATE(#2,200,100):
: RAD=ABS(A-C):: U=A :: V=B
860 IF RAD>20 THEN D=1 ELSE
D=3
870 FOR I=1 TO 360 STEP D
880 A=U+INT(SIN(I*PI/180)*RA
D)
890 B=V+INT(COS(I*PI/180)*RA
D)
900 IF MA=A AND MB=B THEN 92
0 ELSE MA=A :: MB=B
910 GOSUB 1100
920 NEXT I :: ROW=U :: COL=V
 :: GOTO 1370
```

```
930 !****  CHARPLATZ  ******
940 IF F=1 THEN F=2 :: CALL
GCHAR(ROW/8,COL/8+1,H):: CAL
L HCHAR(15,29,H):: GOTO 1370
950 CALL GCHAR(ROW/8,COL/8+1
,A):: IF A=H THEN CALL HCHAR
(15,29,39):: F=1 :: GOTO 137
0 ELSE CALL HCHAR(ROW/8,COL/
8+1,H):: GOTO 1370
960 !***  CALL DRAW ***
970 U=D-B :: V=C-A
980 D1X=SGN(V):: D1Y=SGN(U)
990 D2X=0 :: D2Y=SGN(V)
1000 M=ABS(V):: N=ABS(U)
1010 IF M>N THEN 1040
1020 D2Y=0 :: D2X=SGN(U)
1030 M=MAX(ABS(U),ABS(V))::
N=MIN(ABS(U),ABS(V))
1040 S=INT(M/2):: FOR I=0 TO
 M
1050 GOSUB 1100
1060 S=S+N
1070 IF S<M THEN A=A+D2Y ::
B=B+D2X :: GOTO 1090
1080 S=S-M :: A=A+D1X :: B=B
+D1Y
1090 NEXT I :: CALL LOCATE(#
2,200,100):: GOTO 1370
1100 !*********************
     ***** ++PLOTTEN++ ****
     *********************
1110 XS=B/8 :: XIS=INT(XS)::
 X=(XS-XIS)*8 :: IF XS=XIS T
HEN X=8
1120 YS=A/8 :: YIS=INT(YS)::
 Y=((YS-YIS)*8)*2 :: IF YS=Y
IS THEN Y=16
1130 CX=YIS+1 :: IF YS=YIS T
HEN CX=CX-1
1140 CY=XIS+1 :: IF XS=XIS T
HEN CY=CY-1
1150 IF CY<3 OR CY>26 THEN R
ETURN
1160 IF (CX=CXM AND CY=CYM)=
-1 THEN 1210
1170 CALL GCHAR(CX,CY,E)
1180 IF E>31 AND E<36 THEN W
=1 :: RETURN
1190 IF E=31 THEN NR=NR+1 ::
 E=NR :: IF NR>143 THEN 1740
 ELSE GOSUB 1640 :: NC=NR ::
 CALL HCHAR(CX,CY,NC)ELSE NC
=E
1200 CALL CHARPAT(NC,Y$)
1210 CXM=CX :: CYM=CY
1220 IF X<5 THEN Y=Y-1
1230 Q=ASC(SEG$(Y$,Y,1)):: I
F Q>57 THEN Q=Q-55 ELSE Q=Q-
48
1240 X$=SEG$(C$,1+Q*4,4)
1250 IF X>4 THEN X=X-4
1260 ON X GOTO 1270,1290,129
0,1280
1270 X$="1"&SEG$(X$,2,3):: G
OTO 1300
1280 X$=SEG$(X$,1,3)&"1" ::
GOTO 1300
1290 X$=SEG$(X$,1,X-1)&"1"&S
EG$(X$,X+1,4-X)
1300 Q=.5 :: L=0 :: FOR O=4
TO 1 STEP -1 :: IF SEG$(X$,O
,1)="1" THEN L=L+Q*2
1310 Q=Q*2 :: NEXT O
1320 IF L>9 THEN X$=CHR$(55+
L)ELSE X$=CHR$(L+48)
1330 Y$=SEG$(Y$,1,Y-1)&X$&SE
G$(Y$,Y+1,16-Y)
1340 CALL CHAR(NC,Y$)
1350 RETURN
1360 !*********************
     *****  GRIFFEL  ******
1370 CALL KEY(0,Q,S):: IF S<
>0 THEN CALL SOUND(-100,440,
10):: CALL COLOR(0,6,1):: GO
TO 1440 ELSE CALL COLOR(0,3.
5+PO,1):: PO=-PO
1380 ST=PRS :: CALL JOYST(1,
X,Y):: CALL KEY(1,Q,S):: IF
Q<>18 THEN 1400
1390 IF X<>0 OR Y<>0 THEN ST
=.5 ELSE CALL SOUND(-100,880
,10):: ON PR GOTO 720,660,84
0,820,940
1400 IF Y=4 AND ROW>20 THEN
ROW=ROW-Y/ST ELSE IF Y=-4 AN
D ROW<162 THEN ROW=ROW-Y/ST
1410 IF X=-4 AND COL>28 THEN
 COL=COL+X/ST ELSE IF X=4 AN
D COL<208 THEN COL=COL+X/ST
1420 CALL LOCATE(#W,(ROW-7)+
RA,COL+CA):: GOTO 1370
1430 !*********************
     **** obtionen  ******
1440 RA,CA=0
1450 CALL KEY(0,K,S):: IF S=
0 OR S=-1 THEN CALL COLOR(0,
3.5+PO,1):: PO=-PO :: GOTO 1
450
1460 X=0 :: IF K=101 THEN Y=
4 :: GOTO 1400 ELSE IF K=120
 THEN Y=-4 :: GOTO 1400 ELSE
 Y=0 :: IF K=115 THEN X=-4 :
: GOTO 1400 ELSE IF K=100 TH
EN X=4 :: GOTO 1400
1470 IF K=13 THEN CALL SOUND
(-100,131,10,196,15):: GOTO
1510
1480 IF K<>32 THEN 1370 ELSE
 CALL HCHAR(PR*3+1,28,31)::
CALL LOCATE(#2,200,100):: CA
LL HCHAR(15,29,39):: W,F=1
```

```
1490 IF K=32 THEN PR=PR+1 ::
 IF PR>7 THEN PR=1
1500 CALL HCHAR(PR*3+1,28,30
):: GOTO 1440
1510 ON PR GOTO 1520,1530,15
40,1550,1560,1570,1600
1520 GOSUB 1610 :: ROW=ROW+4
 :: COL=COL+4 :: CALL SPRITE
(#1,42,8,ROW,COL):: RA=3 ::
CA=-4 :: PRS=1 :: GOTO 1370
1530 GOSUB 1610 :: CALL PATT
ERN(#1,38):: RA=-1 :: PRS=.5
 :: GOTO 1370

1540 CALL SPRITE(#1,36,6,ROW
-7,COL):: PRS=4 :: GOTO 1370
1550 CALL SPRITE(#1,36,6,ROW
-7,COL):: PRS=4 :: GOTO 1370
1560 GOSUB 1610 :: CALL SPRI
TE(#1,38,5,ROW,COL):: RA=-1
:: PRS=.5 :: GOTO 1370
1570 FOR A=2 TO 22 :: FOR B=
3 TO 26 :: CALL GCHAR(A,B,C)
:: IF C=31 THEN 1590 ELSE IF
 C=NR+1 THEN A=22 :: B=26 ::
 GOTO 1590
1580 CALL CHARPAT(C,X$):: CH
$(C-44)=X$&"/"&STR$(A)&"-"&S
TR$(B)
1590 NEXT B :: NEXT A :: GOT
O 1700

1600 FOR A=2 TO 22 :: CALL H
CHAR(A,3,31,24):: NEXT A ::
FOR A=45 TO 143 :: CALL CHAR
(A,"0"):: NEXT A :: NR=44 ::
 GOSUB 1640 :: GOTO 1440
1610 ROW=INT(ROW/8)*8+1 :: C
OL=INT(COL/8)*8+1 :: RETURN
1620 !***** CHAR ANZEIGE ***
1630 !*** DISPLAY *****
1640 IF LEN(STR$(143-NR))=2
THEN 1660
1650 CALL CHAR(43,Z$(143-NR)
):: CN=31 :: GOTO 1680
1660 X$=STR$(143-NR):: CALL
CHAR(44,Z$(ASC(SEG$(X$,1,1))
-48))
1670 CN=44 :: CALL CHAR(43,Z
$(ASC(SEG$(X$,2,1))-48))
1680 CALL HCHAR(18,30,43)::
CALL HCHAR(18,29,CN):: RETUR
N
1690 !****  AUSDRUCKEN  ****
1700 CALL CHARSET :: CALL DE
LSPRITE(ALL):: CALL CLEAR ::
 CALL SCREEN(8)
1710 FOR I=1 TO 99 :: PRINT
I+44;CH$(I): :
1720 CALL KEY(0,K,S):: IF S=
0 THEN 1720
1730 NEXT I :: GOTO 1710
1740 F,W=1 :: NR=NR-1 :: CAL
L LOCATE(#2,200,100):: GOTO
1370
1750 !****  ERKLAERUNG  ****
1760 FOR I=65 TO 90 :: CALL
CHARPAT(I,X$):: CALL CHAR(I+
32,X$):: NEXT I :: FOR I=9 T
O 12 :: CALL COLOR(I,2,4)::
NEXT I
1770 CALL CHAR(123,"00000402
FF0204",125,"00000000FC",126
,"0")
1780 CALL SPRITE(#1,123,16,2
00,205):: F=7.5
1790 READ E,B :: IF E=1000 T
HEN 1890
1800 FOR A=1 TO B :: READ C,
X$
1810 FOR I=1 TO LEN(X$):: IF
 SEG$(X$,I,1)=" " THEN D=31
ELSE D=ASC(SEG$(X$,I,1))
1820 CALL HCHAR(C,3+I,D):: N
EXT I :: NEXT A :: CALL LOCA
TE(#1,E,205)
1830 DISPLAY AT(24,4)SIZE(20
):RPT$(CHR$(31),20)
1840 CALL KEY(0,K,S):: IF S<
>0 THEN GOSUB 1880 :: GOTO 1
790
1850 CALL COLOR(#1,8.5+F)::
F=-F
1860 FOR I=1 TO 50 :: NEXT I
 :: DISPLAY AT(24,4)SIZE(20)
:"bitte";"~";"taste";"~";"dr
uecken" :: CALL COLOR(#1,8.5
+F):: F=-F
1870 FOR I=1 TO 100 :: NEXT
I :: GOTO 1830
1880 FOR I=2 TO 22 :: CALL H
CHAR(I,4,31,23):: NEXT I ::
RETURN
1890 F=1 :: CALL DELSPRITE(#
1):: FOR I=9 TO 12 :: CALL C
OLOR(I,4,1):: NEXT I :: GOTO
 620
1900 !GP+
1910 DATA 200,10,4,"......so
ft}plot",6,"soft}plot DIENT
ZUR",8,"EINFACHEN ERSTELLUNG
",10,"EINFARBIGER GRAPHIKEN.
"

1920 DATA 12,"ES KOENNEN INS
GESAMMT",14,"99 ZEICHEN =633
4 DOTS",16,"GESETZT WERDEN.
DAMIT",18,"SIE soft}plot SIN
NVOLL"
1930 DATA 20,"ANWENDEN,BEACH
TEN SIE",22,"BITTE FOLGENDE
PUNKTE:"
1940 DATA 200,10,3,"DURCH DR
UECKEN VON",5,"space KOENNEN
```

```
 SIE",7,"ZWISCHEN DEN VERSCH
IE-",9,"DENEN OBTIONEN WAEHL
EN.",11,"DRUECKEN VON enter"
1950 DATA 13,"FIXIERT DIE GE
WUENSCHTE",15,"FUNKTION.DER
BLINKENDE",17,"CURSOR ZEIGT
IHNEN,",19,"WELCHE OPTION GE
RADE"
1960 DATA 21,"AUSGEFUEHRT WI
RD."
1970 DATA 200,10,3,"joystick
[1]:",5,"PUNKTWEISE BEWEGUNG
 DES",7,"ZEICHENSTIFTES. UM
DEN",9,"STIFT SCHNELLER ZU B
E-",11,"WEGEN,JOYSTICK IN DI
E"

1980 DATA 13,"GEWUENSCHTE RI
CHTUNG  ",15,"DANN button [=
SCHUSS-",17,"KNOPF] DRUECKEN
.",19, pfeiltasten: PUNKTWEI
SE",21,"BEWEGUNG DES STIFTES
."
1990 DATA 17,10,3,"radieren:
",5,"DURCH DRUECKEN DES",7,"
button WIRD DAS 4/4",9,"DOT
GROSSE QUADRAT",11,"UNTER DE
M RADIERGUMMI"
2000 DATA 13,"GELOESCHT UND
KANN",15,"NUN NEU GEPLOTTET"
,17,"WERDEN.",19,"RUECKSPRUN
G ZUM MENUE",21,"DURCH space
."
2010 DATA 41,10,3,"vergroess
ern:",5,"DAS UNTER DEM QUADR
AT",7,"LIEGENDE ZEICHEN WIRD
",9,"NACH DRUECKEN DES BU-",
11,"TON IM UNTERSTEN"
2020 DATA 13,"FELD VERGROESS
ERT",15,"DARGESTELLT. UM DAS
",17,"QUADRAT WEITERBEWEGEN"
,19,"ZU KOENNEN,DEN BUTTON",
21,"NOCHMAL DRUECKEN."
2030 DATA 65,10,3,"kreis]fun
ktion:",5,"BUTTON DRUECKEN,U
M DEN",7,"MITTELPUNKT DES KR
EISES",9,"FESTZULEGEN.STIFT
NACH",11,"OBEN/UNTEN BEWEGEN
,UM"
2040 DATA 13,"DEN RADIUS FES
TZULEGEN.",15,"DANN NOCHMAL
DEN BUTTON",17,"DRUECKEN,UND
 DAS PRO-",19,"GRAMM PLOTTET
 DEN",21,"GEWUENSCHTEN KREIS
."
2050 DATA 59,10,3,"linie]fun
ktion:",5,"BUTTON DRUECKEN,U
M DEN",7,"ANF.PUNKT FESTZULE
GEN,",9,"DANN STIFT 2 ZUM EN
D-",11,"PUNKT FUEHREN UND DE
N"
2060 DATA 13,"BUTTON DRUECKE
N.",15,"laengeres HALTEN DES
",17,"BUTTON PLOTTET EINEN",
19,"PUNKT. DIES WIRD DURCH",
21,"2 TOENE ANGEZEIGT."
2070 DATA 113,19,3,"kopier]f
unktion:",5,"DIENT ZUM SCHNE
LLEN",6,"SETZEN VON GLEICHAR
TI-",7,"GEN ZEICHEN.DAS QUAD
RAT",8,"UEBER DAS ZEICHEN FU
E-"
2080 DATA 9,"HREN UND DURCH
BUTTON",10,"SPEICHERN.DRUECK
EN DES",11,"BUTTON KOPIERT E
S NUN",12,"AN JEDER GEWUENSC
HTEN",13,"STELLE. zeichentau
sch:"

2090 DATA 14,"ZEICHEN SETZEN
 - DANN",15,"WIEDER BUTTON.N
UN KANN",16,"EIN NEUES ZEICH
EN",17,"AUFGENOMMEN WERDEN."
,18,"vorsicht:WIRD EIN MEHR-
"

2100 DATA 19,"FACH KOPIERTES
 ZEICHEN",20,"NACHGEPLOTTET,
AENDERN",21,"SICH DAMIT AUCH
 ALLE",22,"KOPIEN DES ZEICHE
NS."
2110 DATA 137,10,3,"zeichen]
ausdruck:",5,"UM EINE GRAPHI
K AUCH",7,"IN ANDEREN PROGRA
MMEN",9,"VERWENDEN ZU KOENNE
N,",11,"KANN MAN SICH DEN"
2120 DATA 13,"CHARAKTERCODE
UND DIE",15,"POSITION DER ZE
ICHEN",17,"MIT space AUFLIST
EN .",19,"DAMIT IST DAS PROG
RAMM",21,"JEDOCH beendet."
2130 DATA 161,7,3,"loesch]fu
nktion:",5,"MIT DIESER OBTIO
N",7,"KOENNEN SIE DEN BILD-"
,9,"SCHIRM LOESCHEN. DER"
2140 DATA 11,"ZEICHENVORRAT
STEHT",13,"DANNACH WIEDER vo
11",15,"ZUR VERFUEGUNG."
2150 DATA 200,9,5," -ZUM SCH
LUSS EIN TIP-",7,"MACHEN SIE
 SICH SCHRITT",9,"FUER SCHRI
TT MIT DEN",11,"FUNKTIONEN D
ES PRO-",13,"GRAMMS VERTRAUT
,DENN"
2160 DATA 15,"NUR SO WERDEN
SIE DEN",17,"OPTIMALEN NUTZE
N AUS",19,"DEM PROGRAMM ZIEH
EN.",22,"......viel-spass"
2170 DATA 1000,1000
```

ACHTUNG!
089/1298011 Ab 1.10.85!

MATHE TABELLEN TORE

Tabellenberechnung

Nach Erscheinen des Titelbilds wird nach dem Tagesdatum sowie der Benutzung eines Druckers gefragt. Hier besteht die Möglichkeit, bei seriellen Druckern die Bezeichnung RS232 ... (weitere Bez. je nach Typ), oder bei parallelen Druckern die Bezeichnung PIO ... (weitere Bez. je nach Typ) anzugeben.

Im Menue sind folgende Optionen vorgesehen: — Arbeitsblatt — Daten einlesen — Hardcopy — Daten speichern — Berechnung nach Formeln — Help-Funktion

Hier wird auch die Dateibezeichnung sowie das bei Eröffnung eingegebene Tagesdatum angezeigt.

Arbeitsblatt: Es können max. 10 Textbezeichnungen sowie 10 Werte erfaßt werden. Sollte ein Wert nicht unmittelbar unter den nächsten geschrieben werden, so ist in dieser Wert-Spalte, die übersprungen werden soll, eine „0" (Null) einzugeben. Wenn dies nicht geschieht, werden die nachfolgenden Werte nicht bearbeitet bzw. berücksichtigt.

Bei Text ist dies unerheblich.

Am unteren Bildschirmende wird die Gesamtsumme ausgeworfen. Außerdem wird während des Dialogs permanent die Möglichkeit der weiteren Vorgehensweise angezeigt.

Es können jederzeit Änderungen im Text oder im Wertbereich vorgenommen werden. Nach Eingabe bzw. Änderung können Texte und Werte (der Übersichtlichkeit halber) mit der Funktion „T" tabellarisch aufgelistet werden.

Mit Funktion „P" werden die eingegebenen Werte prozentual pro Position ausgewertet. Diese können wiederum mit der Funktion „G" graphisch dargestellt werden.

Hierbei werden allerdings nur die positiven Werte angezeigt (+. Beträge). Durch nochmaliges Drücken der Taste „t" wird wieder die Tabelle aufgelistet. Funktion „C" steht für Hardcopy, die ebenfalls im Arbeitsblatt möglich ist. Graphikausdruck ist allerdings auf Grund der begrenzten Speicherkapazität nicht vorgesehen.

Die Funktion „F" ermöglicht die Berechnung nach Formeln (bereits integriert).

Durch Drücken der Taste „M" wird der Mittelwert der eingegebenen Werte ausgeworfen.

Daten einlesen: Es können mit diesem Programm bereits erstellte Daten eingelesen und bearbeitet werden. Der bei der Eröffnung angegebene Dateiname sowie das Tagesdatum wird mit dem Einspielen der neuen Datei automatisch überschrieben und auf neuen Stand gebracht. Nach Beendigung des Ladevorgangs verzweigt das Programm zum Menue und die neu geladene Datei sowie das Erstellungsdatum wird angezeigt.

Sollten beim Ladevorgang Fehler auftreten (File oder Data-Error) so wird dies durch eine Mitteilung angezeigt. Soll trotzdem eine neue Datei eingelesen werden, so ist das Programm abzubrechen und wieder neu zu starten.

Daten abspeichern: Es werden nur die im Arbeitsblatt erstellten Werte und Texte gesichert, nicht aber die prozentualen oder graphischen Auswertungen.

Help-Funktion: Mit dieser Funktion kann jederzeit, im Menue oder im Arbeitsblatt, festgestellt werden, welche Bedeutung die angegebenen Parameter haben.

Gerhard Hübner

Torschießen

Torschießen ist ein Spiel für zwei Spieler. Der eine Spieler schießt und der andere wehrt ab. Für jedes Tor oder gehaltenen Ball gibt es einen Punkt. Wer zuerst 10 Punkte hat, hat gewonnen.

Steuerung: siehe Programm.

Achtung:

1. Berühren sich die Spieler, so wird „FOUL" angezeigt und der Angriff wiederholt.
2. Kommt der Torwart zu weit aus seinem Tor, kann er sich nur ganz langsam bewegen.

Programmerläuterungen:
Hauptprogramm 220—400
Sub Punkte 410—470 (Spieler)
Sub Gepunk 480—540 (Torwart)
Sub Gehalten 550—610
Sub Stand 620—649 (Spielstand)
Sub Schuß 650—750
Sub Foul 760—800
Sub Bildschirm 810—1080 (Graphik)
Sub Spielanleitung 1090—1300
Sub Torwart 1310—1360 (Torwartsteuerung)
Sprite 1 Spieler
Sprite 2 Torwart
Sprite 3 Ball
Empfindlichkeit des Torwarts ist in den „Call Coinc(...)"-Anweisungen festgelegt (Zeile 280,340, 600).

```
100 ! **********************
110 ! *                    *
120 ! *   TORESCHIESSEN    *
130 ! *                    *
140 ! *   Copyright  by    *
150 ! *                    *
160 ! *   Steffen Trefz    *
170 ! *                    *
180 ! *                    *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole  *
210 ! *     Ext. Basic     *
220 ! *   Joystick (1&2)   *
230 ! *                    *
260 ! *  Speicherbelegung  *
270 ! *     5610 Bytes     *
280 ! *                    *
290 ! **********************
300 CALL CLEAR :: CALL SPIEL
ANLEITUNG :: CALL BILDSCHIRM
310 CALL SPRITE(#1,124,11,16
5,90):: CALL SPRITE(#2,112,1
6,165,125):: CALL SPRITE(#3,
136,8,96,130)
320 CALL JOYST(1,X,Y):: CALL
 TORWART :: CALL KEY(0,K,S):
: IF K=32 THEN CALL DELSPRIT
E(ALL):: CALL CLEAR :: END
330 CALL MOTION(#1,-Y*2,X*2)
340 CALL COINC(#1,#2,6,T)::
CALL COINC(#2,#3,10,P):: CAL
L COINC(#1,#3,6,Q)
350 IF Q=-1 THEN CALL FOUL :
: GOTO 310 ELSE 360
360 IF P=-1 THEN CALL GEPUNK
(G,W):: CALL GEHALTEN(G,W)::
 GOTO 460 ELSE 370
370 IF T=-1 THEN CALL SCHUSS
 :: GOTO 320 ELSE 380
380 CALL POSITION(#2,V,H)
390 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,-Y*2,X*2):: CALL
TORWART
400 CALL COINC(#2,#3,10,U)::
 CALL COINC(#1,#3,6,F)
410 IF F=-1 THEN CALL FOUL :
: GOTO 310 ELSE IF U=-1 THEN
 CALL GEPUNK(G,W):: CALL GEH
ALTEN(G,W):: GOTO 460 ELSE 4
20
420 CALL POSITION(#2,V,H)
430 IF U=0 THEN IF V<90 AND
V>20 THEN IF H>89 AND H<157
THEN CALL PUNKTE(G,W):: GOTO
 450 ELSE 440
440 IF V<89 OR H>250 OR H<10
 THEN CALL DELSPRITE(#2):: C
ALL HCHAR(2,12,40,10):: DISP
LAY AT(2,12)SIZE(6):"VORBEI"
 :: GOTO 450 ELSE 320
450 CALL MOTION(#1,0,0,#2,0,
0,#3,0,0):: CALL STAND(G,W):
: FOR I=1 TO 250 :: NEXT I :
: GOTO 310
460 CALL POSITION(#2,R,S)::
IF R>=180 THEN 310 ELSE CALL
 STAND(G,W):: GOTO 320
470 SUB PUNKTE(G,W)
480 CALL SOUND(100,523,2,659
,2):: CALL HCHAR(2,12,40,10)
490 G=G+1 :: DISPLAY AT(2,12
)SIZE(3):"TOR" :: IF G=10 TH
EN DISPLAY AT(2,12)SIZE(4):"
ENDE" :: GOTO 500 ELSE CALL
STAND(G,W):: SUBEXIT
500 CALL KEY(0,K,S):: IF K=3
2 THEN CALL DELSPRITE(ALL)::
 CALL CLEAR :: END
510 CALL KEY(2,K,S):: CALL M
OTION(#1,0,0,#2,0,0,#3,0,0):
: IF K=18 THEN G,W=0 :: DISP
LAY AT(2,12)SIZE(5):"START"
:: SUBEXIT ELSE 500
520 SUBEND
530 SUB GEPUNK(G,W)
540 CALL SOUND(100,261,2,326
,2):: CALL HCHAR(2,12,40,10)
550 W=W+1 :: IF W=10 THEN DI
SPLAY AT(2,12)SIZE(4):"ENDE"
 :: GOTO 560 ELSE CALL STAND
(G,W):: SUBEXIT
560 CALL KEY(0,K,S):: IF K=3
2 THEN CALL DELSPRITE(ALL)::
 CALL CLEAR :: END
570 CALL KEY(1,K,S):: CALL M
OTION(#1,0,0,#2,0,0,#3,0,0):
: IF K=18 THEN G,W=0 :: DISP
LAY AT(2,12)SIZE(5):"START"
:: SUBEXIT ELSE 560
580 SUBEND
590 SUB GEHALTEN(G,W)
600 A=16 :: B=3 :: CALL PATT
ERN(#1,124):: CALL MOTION(#2
,A,B,#3,0,0,#1,0,0)
610 DISPLAY AT(2,12)SIZE(8):
"GEHALTEN" :: CALL STAND(G,W
)
620 FOR I=1 TO 200 :: NEXT I
 :: CALL MOTION(#2,0,0)
630 CALL POSITION(#2,V,H)::
IF V>=180 THEN SUBEXIT ELSE
640
640 CALL COINC(#2,#3,10,U)::
 IF U=-1 THEN 600 ELSE SUBEX
IT
650 SUBEND
660 SUB STAND(G,W)
670 DISPLAY AT(4,12)SIZE(1):
CHR$(G+48):: DISPLAY AT(4,15
)SIZE(1):":" :: DISPLAY AT(4
,18)SIZE(1):CHR$(W+48)
```

```
680 SUBEND
690 SUB SCHUSS
700 CALL KEY(1,K,S):: CALL J
OYST(1,X,Y):: CALL TORWART !
:: CALL JOYST(2,X1,Y1)
710 CALL MOTION(#1,-Y*2,X*2)
:: CALL MOTION(#2,-Y*2,X*2)
720 CALL COINC(#1,#3,6,Q)::
IF Q=-1 THEN CALL FOUL :: SU
BEXIT ELSE 730
730 IF K=18 AND Y=0 AND X=4
THEN N=-20 :: L=15 :: GOTO 7
80 :: SUBEXIT ELSE 740
740 IF K=18 AND Y=0 AND X=-4
 THEN N=-20 :: L=-15 :: GOTO
 780 :: SUBEXIT ELSE 750
750 IF K=18 AND Y=4 AND X=4
THEN N=-20 :: L=8 :: GOTO 78
0 :: SUBEXIT ELSE 760
760 IF K=18 AND Y=4 AND X=-4
 THEN N=-20 :: L=-8 :: GOTO
780 :: SUBEXIT ELSE 770
770 IF K=18 AND Y=4 AND X=0
THEN N=-20 :: L=0 :: GOTO 78
0 :: SUBEXIT ELSE 700
780 CALL PATTERN(#1,120):: C
ALL MOTION(#2,N,L)
790 SUBEND
800 SUB FOUL
810 CALL COINC(#1,#3,6,Q)::
IF Q=-1 THEN CALL HCHAR(2,12
,40,10):: DISPLAY AT(2,12)SI
ZE(4):"FOUL" :: GOTO 820
820 CALL DELSPRITE(#3):: CAL
L MOTION(#2,0,0,#1,0,0):: CA
LL PATTERN(#1,140)
830 FOR I=1 TO 220 :: NEXT I
 :: CALL DELSPRITE(ALL):: SU
BEXIT
840 SUBEND
850 SUB BILDSCHIRM
860 CALL SCREEN(2):: CALL CH
AR(33,"3C3C99A5A55A3C3C",40,
RPT$("F",16),104,RPT$("F",16
))
870 CALL CHAR(96,"0707070707
0707FF",97,RPT$("07",9),98,R
PT$("E0",8),99,"E0E0E0E0E0E0
E0FF",100,"FFFFFFFF")
880 CALL CHAR(116,"000000000
00000FF",102,"FFFFFFFFE0E0E0
E0",103,"FFFFFFFF07070707")
890 CALL CHAR(120,"030303010
3070F1B3303030303030307C0C0C
080C0E0F0DCC0C0E07030303800"
)
900 CALL CHAR(124,"030303011
F1F1B1B0303033F3F2000080808
000C0C0F8F880F0F0303030380")
910 CALL CHAR(112,RPT$("0",5
2)&"183C3C18")
920 CALL CHAR(136,"101313131
11F0F070707070606060E10909
09010F0E0C0C0C0C0C0C0C0C0E")
930 CALL CHAR(140,"38BABA93F
F3938397B7F1F3F266C181000000
0C0C0F8A0C080E0B09080C")
940 CALL HCHAR(8,15,100,7)::
 CALL VCHAR(9,14,98,3):: CAL
L VCHAR(9,21,97,3)
950 CALL HCHAR(8,14,102):: C
ALL HCHAR(8,21,103)
960 CALL VCHAR(12,14,99):: C
ALL VCHAR(12,21,96)
970 CALL HCHAR(12,1,116,13):
: CALL HCHAR(12,15,116,6)::
CALL HCHAR(12,22,116,11)
980 CALL HCHAR(1,1,33,43)
990 FOR I=12 TO 23 :: FOR J=
2 TO 5 :: CALL HCHAR(J,I,40)
:: NEXT J :: NEXT I
1000 FOR I=2 TO 5 :: CALL HC
HAR(I,24,33,20):: NEXT I
1010 CALL HCHAR(6,12,33,54)
1020 FOR I=8 TO 11 :: FOR J=
1 TO 13 :: CALL HCHAR(I,J,10
4):: NEXT J :: NEXT I
1030 FOR I=8 TO 10 :: FOR J=
22 TO 32 :: CALL HCHAR(I,J,1
04):: NEXT J :: NEXT I
1040 FOR J=1 TO 13 :: CALL H
CHAR(11,J,104):: FOR K=22 TO
 32 :: CALL HCHAR(11,K,104):
: NEXT K :: NEXT J
1050 FOR I=9 TO 11 :: FOR J=
15 TO 20 :: CALL HCHAR(I,J,1
04):: NEXT J :: NEXT I
1060 ! COLORS *************
1070 CALL COLOR(2,11,11,8,16
,1,9,16,10,10,10,10,11,16,10
)
1080 FOR I=3 TO 7 :: CALL CO
LOR(I,2,11):: NEXT I
1090 CALL MAGNIFY(3):: CALL
SCREEN(13)
1100 SUBEND
1110 SUB SPIELANLEITUNG
1120 CALL CLEAR :: DISPLAY A
T(10,4):"WUENSCHEN SIE EINE
SPIELAN- LEITUNG ?(J/N)"
1130 CALL KEY(0,K,S):: IF S=
0 THEN 1130
1140 IF K=74 OR K=106 THEN 1
150 ELSE SUBEXIT
1150 CALL CLEAR :: CALL SCRE
EN(8)
1160 PRINT "******* TORSCHIE
SSEN *******" :: PRINT :::::
;
1170 PRINT "SPIELERANLEITUNG
:" :: PRINT "STEUERUNG ERFOL
```

```
CT UEBER.....JOYSTICK-1"
1180 PRINT "DER SCHUSS WIRD
DURCH KNOPF-DRUCK UND DRUECK
EN IN ENT-..SPRECHENDE RICHT
UNG AUSGE-..LOEST !"
1190 PRINT ::::::: PRINT "BI
TTE EINE TASTE DRUECKEN"
1200 CALL KEY(0,K,S):: IF S=
0 THEN 1200
1210 CALL CLEAR
1220 PRINT "TORWARTANLEITUNG
:"
1230 PRINT "STEUERUNG ERFOLG
T UEBER.....JOYSTICK-2"
1240 PRINT "DER TORWART KANN
 DIE SCHUES-SE ABWEHREN UND
3? PUNKTEN"
1250 PRINT ::::::::::: PRINT
 "BITTE EINE TASTE DRUECKEN"
1260 CALL KEY (0,K,S):: IF S=
0 THEN 1260
1270 CALL CLEAR
1280 PRINT " NEUER START AUF
 KNOPFDRUCK..DES VERLIERERS"
:::::::: PRINT " ENDE AUF LE
ERTASTE"::::::: PRINT " ALF
HALSCH HERAUSNEHMEN!"::::::
:
1290 PRINT " BITTE EINE TAST
E DRUECKEN"
1300 CALL KEY(0,K,S):: IF S=
0 THEN 1300 ELSE CALL CLEAR
:: SUBEXIT
1310 SUBEND
1320 SUB TORWART
1330 CALL JOYST(2,X1,Y1)
1340 CALL MOTION(#9,-Y1*2,X1
*2)
1350 CALL POSITION(#3,R,S)
1360 IF R>=140 THEN CALL MOT
ION(#3,0,0)
1370 SUBEND
```



```
100 ! ***********************
110 ! *                     *
120 ! *TABELLENBERECHNUNGEN*
130 ! *                     *
140 ! *    Copyright by     *
150 ! *                     *
160 ! *  Peter Eisenbraun   *
170 ! *                     *
190 ! * Benoetigte Geraete *
200 ! *  TI99/4A  Konsole  *
210 ! *     Ext. Basic     *
220 ! *                     *
260 ! *  Speicherbelegung  *
270 ! *    11184 Bytes     *
280 ! *                     *
290 ! ***********************
295 !
300 CALL SCREEN(2):: FOR T=9
 TO 12 :: CALL COLOR(T,16,5)
:: NEXT T :: FOR I=5 TO 8 ::
 CALL COLOR(I,8,2):: NEXT I
310 FOR U=0 TO 4 :: CALL COL
OR(U,11,2):: NEXT U :: DISPL
AY AT(5,3)ERASE ALL:"   TABE
LLENBERECHNUG    " :: FOR DL=
1 TO 500 :: NEXT DL
320 ON WARNING NEXT :: DISPL
AY AT(3,10)ERASE ALL:"TT.MM.
JJ" :: DISPLAY AT(9,10):"DAT
EINAME" :: ACCEPT AT(5,10)SI
ZE(2)VALIDATE(DIGIT):TT
330 ACCEPT AT(5,13)SIZE(2)VA
LIDATE(DIGIT):MM :: ACCEPT A
T(5,16)SIZE(2)VALIDATE(DIGIT
):JJ
340 IF TT<1 OR TT>31 OR MM<1
 OR MM>12 THEN 320 ELSE ACCE
PT AT(11,10)SIZE(10)VALIDATE
(UALPHA,"."):DN$
350 DISPLAY AT(15,10):"DRUCK
ER J N" :: ACCEPT VALIDATE("
JN"):EG$ :: IF EG$="J" THEN
DISPLAY AT(17,1):"BEZEICHNUN
G ?"
360 IF EG$="N" THEN 370 :: D
ISPLAY AT(19,1):"RS232.BA=..
.ODER PIO ?" :: ACCEPT AT(23
,1)VALIDATE(UALPHA,".=/-1234
567890"):DR$
370 DISPLAY AT(3,11)ERASE AL
L:"auswahl" :: DISPLAY AT(8,
1):"a>ARBEITSBLATT...r>READ
DATA" :: DISPLAY AT(10,1):"h
>HELP ROUTINE...s>SAVE DATA"
380 DISPLAY AT(12,1):"c>HARD
COPY" :: DISPLAY AT(22,1):"d
atum ";TT;".";MM;".";JJ :: D
ISPLAY AT(24,1):"datei ";TAB
(8);DN$
390 DISPLAY AT(18,1):"drucke
r";TAB(20);DR$ :: CALL KEY(0
```

```
,K,S):: IF S=0 THEN 390 ELSE
 400
400 IF K=65 THEN 410 :: IF K
=67 THEN 1580 :: IF K=83 THE
N 880 :: IF K=82 THEN 920 ::
 IF K=72 THEN 1540 ELSE 370
410 DISPLAY AT(1,1)ERASE ALL
:F1$ :: DISPLAY AT(3,1):F2$
:: DISPLAY AT(5,1):F3$
420 DISPLAY AT(1,16):USING "
######.##":Z1 :: DISPLAY AT(
3,16):USING "######.##":Z2 :
: DISPLAY AT(5,16):USING "##
####.##":Z3
430 DISPLAY AT(7,1):F4$ :: D
ISPLAY AT(9,1):F5$ :: DISPLA
Y AT(11,1):F6$ :: DISPLAY AT
(13,1):F7$
440 DISPLAY AT(7,16):USING "
######.##":Z4 :: DISPLAY AT(
9,16):USING "######.##":Z5 :
: DISPLAY AT(11,16):USING "#
#####.##":Z6
450 DISPLAY AT(13,16):USING
"######.##":Z7
460 DISPLAY AT(15,1):F8$ ::
DISPLAY AT(17,1):F9$ :: DISP
LAY AT(19,1):F10$
470 DISPLAY AT(15,16):USING
"######.##":Z8 :: DISPLAY AT
(17,16):USING "######.##":Z9
 :: DISPLAY AT(19,16):USING
"######.##":Z10
480 DISPLAY AT(1,27):"1" ::
DISPLAY AT(3,27):"2" :: DISP
LAY AT(5,27):"3" :: DISPLAY
AT(7,27):"4" :: DISPLAY AT(9
,27):"5" :: DISPLAY AT(11,27
):"6"
490 DISPLAY AT(13,27):"7" ::
 DISPLAY AT(15,27):"8" :: DI
SPLAY AT(17,27):"9" :: DISPL
AY AT(19,26):"10"
500 DISPLAY AT(22,1):"SUMME
TOTAL" :: ST=Z1+Z2+Z3+Z4+Z5+
Z6+Z7+Z8+Z9+Z10 :: DISPLAY A
T(22,16):USING "######.##":S
T
510 DISPLAY AT(24,3):"~~text
e~aend~einf~j!n~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 510 :
: IF K=74 THEN 540 :: IF K=7
8 THEN 520 ELSE 510
520 DISPLAY AT(24,3):"~~wert
e~aend~einf~j!n~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 520 :
: IF K=74 THEN 670 ELSE 530
530 IF K=78 THEN 780 ELSE 37
0
540 ON WARNING NEXT :: DISPL
AY AT(24,3):"~~welche~poitio
ns~nr~~" :: ACCEPT AT(24,26)
SIZE(2)BEEP VALIDATE(DIGIT):
PS :: GOTO 560
550 DISPLAY AT(24,3):"~~inse
rt~neuen~text~~~" :: RETURN
560 IF PS<1 OR PS>10 THEN 54
0 :: IF PS=1 THEN GOSUB 550
:: ACCEPT AT(1,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F1$ :
: GOTO 510
570 IF PS=2 THEN GOSUB 550 :
: ACCEPT AT(3,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F2$ ::
 GOTO 510
580 IF PS=3 THEN GOSUB 550 :
: ACCEPT AT(5,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F3$ ::
 GOTO 510
590 IF PS=4 THEN GOSUB 550 :
: ACCEPT AT(7,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F4$ ::
 GOTO 510
600 IF PS=5 THEN GOSUB 550 :
: ACCEPT AT(9,1)SIZE(10)BEEP
 VALIDATE(UALPHA,"."):F5$ ::
 GOTO 510
610 IF PS=6 THEN GOSUB 550 :
: ACCEPT AT(11,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F6$ :
: GOTO 510
620 IF PS=7 THEN GOSUB 550 :
: ACCEPT AT(13,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F7$ :
: GOTO 510
630 IF PS=8 THEN GOSUB 550 :
: ACCEPT AT(15,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F8$ :
: GOTO 510
640 IF PS=9 THEN GOSUB 550 :
: ACCEPT AT(17,1)SIZE(10)BEE
P VALIDATE(UALPHA,"."):F9$ :
: GOTO 510
650 IF PS=10 THEN GOSUB 550
:: ACCEPT AT(19,1)SIZE(10)BE
EP VALIDATE(UALPHA,"."):F10$
 :: GOTO 510
660 DISPLAY AT(24,3)SIZE(20)
:"~~insert~neuen~wert~~~" ::
 RETURN
670 ON WARNING NEXT :: DISPL
AY AT(24,5)SIZE(20):"welche~
positions~nr~" :: ACCEPT AT(
24,26)SIZE(2)BEEP VALIDATE(D
IGIT):PP :: GOTO 680
680 IF PP<1 OR PP>10 THEN 67
0 :: IF PP=1 THEN GOSUB 660
:: ACCEPT AT(1,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z1 :: GO
TO 500
690 IF PP=2 THEN GOSUB 660 :
```

```
: ACCEPT AT(3,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z2 :: GOT
O 500
700 IF PP=3 THEN GOSUB 660 :
: ACCEPT AT(5,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z3 :: GOT
O 500
710 IF PP=4 THEN GOSUB 660 :
: ACCEPT AT(7,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z4 :: GOT
O 500
720 IF PP=5 THEN GOSUB 660 :
: ACCEPT AT(9,15)SIZE(9)BEEP
 VALIDATE(NUMERIC):Z5 :: GOT
O 500
730 IF PP=6 THEN GOSUB 660 :
: ACCEPT AT(11,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z6 :: GO
TO 500
740 IF PP=7 THEN GOSUB 660 :
: ACCEPT AT(13,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z7 :: GO
TO 500
750 IF PP=8 THEN GOSUB 660 :
: ACCEPT AT(15,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z8 :: GO
TO 500
760 IF PP=9 THEN GOSUB 660 :
: ACCEPT AT(17,15)SIZE(9)BEE
P VALIDATE(NUMERIC):Z9 :: GO
TO 500
770 IF PP=10 THEN GOSUB 660
:: ACCEPT AT(19,15)SIZE(9)BE
EP VALIDATE(NUMERIC):Z10 ::
GOTO 500
780 DISPLAY AT(24,3):"f~p~m~
g~t~c~e~~help!h~" :: CALL KE
Y(0,K,S):: IF S=0 THEN 780 E
LSE 790
790 IF K=80 THEN 810 :: IF K
=72 THEN 1540 :: IF K=84 THE
N 410 :: IF K=71 THEN 970 ::
 IF K=77 THEN 1620 :: IF K=6
7 THEN 1580
800 IF K=69 THEN 370 :: IF K
=70 THEN 1670 ELSE 510
810 IF ST=0 THEN 820 ELSE 83
0
820 DISPLAY AT(24,3):"keine
werte angelegt" :: CALL KEY(
0,K,S):: IF S=0 THEN 820 ELS
E 510
830 X=100. :: DISPLAY AT(22,
25)SIZE(4):X :: DISPLAY AT(2
3,27)SIZE(1):"%" :: P1=Z1/ST
*X :: DISPLAY AT(1,25)SIZE(4
):P1 :: P2=Z2/ST*X
840 DISPLAY AT(3,25)SIZE(4):
P2 :: P3=Z3/ST*X :: DISPLAY
AT(5,25)SIZE(4):P3 :: P4=Z4/
ST*X :: DISPLAY AT(7,25)SIZE
(4):P4
850 P5=Z5/ST*X :: DISPLAY AT
(9,25)SIZE(4):P5 :: P6=Z6/ST
*X :: DISPLAY AT(11,25)SIZE(
4):P6 :: P7=Z7/ST*X :: DISPL
AY AT(13,25)SIZE(4):P7 :: P8
=Z8/ST*X
860 DISPLAY AT(15,25)SIZE(4)
:P8 :: P9=Z9/ST*X :: DISPLAY
 AT(17,25)SIZE(4):P9 :: P10=
Z10/ST*X :: DISPLAY AT(19,25
)SIZE(4):P10
870 CALL KEY(0,K,S):: IF S=0
 THEN 870 ELSE 510
880 CALL CLEAR :: OPEN #1:"C
S1",INTERNAL,SEQUENTIAL,OUTP
UT,FIXED 192 :: PRINT #1:TT,
MM,JJ :: PRINT #1:DN$
890 PRINT #1:F1$,F2$,F3$,F4$
,F5$,F6$,F7$,F8$,F9$,F10$
900 PRINT #1:Z1,Z2,Z3,Z4,Z5,
Z6,Z7,Z8,Z9,Z10
910 CLOSE #1 :: GOTO 370
920 CALL CLEAR :: ON ERROR 1
600
930 OPEN #1:"CS1",INTERNAL,S
EQUENTIAL,INPUT ,FIXED 192 :
: INPUT #1:TT,MM,JJ :: INPUT
 #1:DN$
940 INPUT #1:F1$,F2$,F3$,F4$
,F5$,F6$,F7$,F8$,F9$,F10$
950 INPUT #1:Z1,Z2,Z3,Z4,Z5,
Z6,Z7,Z8,Z9,Z10
960 CLOSE #1 :: GOTO 370
970 DISPLAY AT(1,1)SIZE(15):
"  5%" :: DISPLAY AT(3,1)SIZ
E(15):" 10%" :: DISPLAY AT(5
,1)SIZE(15):" 20%" :: DISPLA
Y AT(7,1)SIZE(15):" 30%"
980 DISPLAY AT(9,1)SIZE(15):
" 40%"
990 DISPLAY AT(11,1)SIZE(15)
:" 50%" :: DISPLAY AT(13,1)S
IZE(15):" 60%" :: DISPLAY AT
(15,1)SIZE(15):" 70%" :: DIS
PLAY AT(17,1)SIZE(15):" 80%"
1000 DISPLAY AT(19,1)SIZE(15
):" 90%"
1010 DISPLAY AT(21,1)SIZE(15
):"100%" :: DISPLAY AT(22,1)
SIZE(15):"....1234567890" ::
 DISPLAY AT(23,1)SIZE(15):".
.............1"
1020 FOR L=14 TO 14 :: CALL
COLOR(L,5,2):: CALL CHAR(136
,"FFFFFFFFFFFFFFFFFFFF")::
NEXT L
1030 FOR OI=13 TO 13 :: CALL
 COLOR(OI,14,2):: CALL CHAR(
128,"FFFFFFFFFFFFFFFFFF")::
```

```
NEXT OI :: IF P1<1 THEN 108
0
1040 IF P1>1 AND P1<6 THEN A
=1 ELSE IF P1>5 AND P1<11 TH
EN A=3 ELSE IF P1>10 AND P3<
21 THEN A=5 ELSE IF P1>20 AN
D P1<31 THEN A=7
1050 IF P1>30 AND P1<41 THEN
 A=9 ELSE IF P1>40 AND P1<51
 THEN A=11 ELSE IF P1>50 AND
 P1<61 THEN A=13
1060 IF P1>60 AND P1<71 THEN
 A=15 ELSE IF P1>70 AND P1<8
1 THEN A=17 ELSE IF P1>80 AN
D P1<91 THEN A=19 ELSE IF P1
>91 THEN A=21
1070 CALL VCHAR(1,7,136,A)
1080 IF P2<1 THEN 1130
1090 IF P2>1 AND P2<6 THEN A
B=1 ELSE IF P2>5 AND P2<11 T
HEN AB=3 ELSE IF P2>10 AND P
2<21 THEN AB=5 ELSE IF P2>20
 AND P2<31 THEN AB=7
1100 IF P2>30 AND P2<41 THEN
 AB=9 ELSE IF P2>40 AND P2<5
1 THEN AB=11 ELSE IF P2>50 A
ND P2<61 THEN AB=13 ELSE IF
P2>60 AND P2<71 THEN AB=15
1110 IF P2>70 AND P2<81 THEN
 AB=17 ELSE IF P2>80 AND P2<
91 THEN AB=19 ELSE IF P2>91
THEN AB=21
1120 CALL VCHAR(1,8,128,AB)
1130 IF P3<1 THEN 1180
1140 IF P3>1 AND P3<6 THEN A
C=1 ELSE IF P3>5 AND P3<11 T
HEN AC=3 ELSE IF P3>10 AND P
3<21 THEN AC=5 ELSE IF P3>20
 AND P3<31 THEN AC=7
1150 IF P3>30 AND P3<41 THEN
 AC=9 ELSE IF P3>40 AND P3<5
1 THEN AC=11 ELSE IF P3>50 A
ND P7<61 THEN AC=13 ELSE IF
P3>60 AND P3<71 THEN AC=15
1160 IF P3>70 AND P3<81 THEN
 AC=17 ELSE IF P3>80 AND P3<
91 THEN AC=19 ELSE IF P3>91
THEN AC=21
1170 CALL VCHAR(1,9,136,AC)
1180 IF P4<1 THEN 1230
1190 IF P4>1 AND P4<6 THEN A
D=1 ELSE IF P4>5 AND P4<11 T
HEN AD=3 ELSE IF P4>10 AND P
4<21 THEN AD=5 ELSE IF P4>20
 AND P4<31 THEN AD=7
1200 IF P4>30 AND P4<41 THEN
 AD=9 ELSE IF P4>40 AND P4<5
1 THEN AD=11 ELSE IF P4>50 A
ND P4<61 THEN AD=13 ELSE IF
P4>60 AND P4<71 THEN AD=15
1210 IF P4>70 AND P4<81 THEN
 AD=17 ELSE IF P4>80 AND P4<
91 THEN AD=19 ELSE IF P4>91
THEN AD=21
1220 CALL VCHAR(1,10,128,AD)
1230 IF P5<1 THEN 1280
1240 IF P5>1 AND P5<6 THEN A
E=1 ELSE IF P5>5 AND P5<11 T
HEN AE=3 ELSE IF P5>10 AND P
5<21 THEN AE=5 ELSE IF P5>20
 AND P5<31 THEN AE=7
1250 IF P5>30 AND P5<41 THEN
 AE=9 ELSE IF P5>40 AND P5<5
1 THEN AE=11 ELSE IF P5>50 A
ND P5<61 THEN AE=13 ELSE IF
P5>60 AND P5<71 THEN AE=15
1260 IF P5>70 AND P5<81 THEN
 AE=17 ELSE IF P5>80 AND P5<
91 THEN AE=19 ELSE IF P5>91
THEN AE=21
1270 CALL VCHAR(1,11,136,AE)
1280 IF P6<1 THEN 1330
1290 IF P6>1 AND P6<6 THEN A
F=1 ELSE IF P6>5 AND P6<11 T
HEN AF=3 ELSE IF P6>10 AND P
6<21 THEN AF=5 ELSE IF P6>20
 AND P6<31 THEN AF=7
1300 IF P6>30 AND P6<41 THEN
 AF=9 ELSE IF P6>40 AND P6<5
1 THEN AF=11 ELSE IF P6>50 A
ND P6<61 THEN AF=13 ELSE IF
P6>60 AND P6<71 THEN AF=15
1310 IF P6>70 AND P6<81 THEN
 AF=17 ELSE IF P6>80 AND P6<
91 THEN AF=19 ELSE IF P6>91
THEN AF=21
1320 CALL VCHAR(1,12,128,AF)
1330 IF P7<1 THEN 1380
1340 IF P7>1 AND P7<6 THEN A
G=1 ELSE IF P7>5 AND P7<11 T
HEN AG=3 ELSE IF P7>10 AND P
7<21 THEN AG=5 ELSE IF P7>20
 AND P7<31 THEN AG=7
1350 IF P7>30 AND P7<41 THEN
 AG=9 ELSE IF P7>40 AND P7<5
1 THEN AG=11 ELSE IF P7>50 A
ND P7<61 THEN AG=13 ELSE IF
P7>60 AND P7<71 THEN AG=15
1360 IF P7>70 AND P7<81 THEN
 AG=17 ELSE IF P7>80 AND P7<
91 THEN AG=19 ELSE IF P7>91
THEN AG=21
1370 CALL VCHAR(1,13,136,AG)
1380 IF P8<1 THEN 1430
1390 IF P8>1 AND P8<6 THEN A
H=1 ELSE IF P8>5 AND P8<11 T
HEN AH=3 ELSE IF P8>10 AND P
3<21 THEN AH=5 ELSE IF P8>20
 AND P8<31 THEN AH=7
1400 IF P8>30 AND P8<41 THEN
 AH=9 ELSE IF P8>40 AND P8<5
```

```
1 THEN AH=11 ELSE IF P8>50 A
ND P8<61 THEN AH=13 ELSE IF
P8>60 AND P8<71 THEN AH=15
1410 IF P8>70 AND P8<81 THEN
 AH=17 ELSE IF P8>80 AND P8<
91 THEN AH=19 ELSE IF P8>91
THEN AH=21
1420 CALL VCHAR(1,14,128,AH)
1430 IF P9<1 THEN 1480
1440 IF P9>1 AND P9<6 THEN A
I=1 ELSE IF P9>5 AND P9<11 T
HEN AI=3 ELSE IF P9>10 AND P
9<21 THEN AI=5 ELSE IF P9>20
 AND P9<31 THEN AI=7
1450 IF P9>30 AND P9<41 THEN
 AI=9 ELSE IF P9>40 AND P9<5
1 THEN AI=11 ELSE IF P9>50 A
ND P9<61 THEN AI=13 ELSE IF
P9>60 AND P9<71 THEN AI=15
1460 IF P9>70 AND P9<81 THEN
 AI=17 ELSE IF P9>80 AND P9<
91 THEN AI=19 ELSE IF P9>91
THEN AI=21
1470 CALL VCHAR(1,15,136,AI)
1480 IF P10<1 THEN 1530
1490 IF P10>1 AND P10<6 THEN
 AJ=1 ELSE IF P10>5 AND P10<
11 THEN AJ=3 ELSE IF P10>10
AND P10<21 THEN AJ=5 ELSE IF
 P10>20 AND P10<31 THEN AJ=7
1500 IF P10>30 AND P10<41 TH
EN AJ=9 ELSE IF P10>40 AND P
10<51 THEN AJ=11 ELSE IF P10
>50 AND P10<61 THEN AJ=13
1510 IF P10>60 AND P10<71 TH
EN AJ=15 ELSE IF P10>70 AND
P10<81 THEN AJ=17 ELSE IF P1
0>80 AND P10<91 THEN AJ=19 E
LSE IF P10>91 THEN AJ=21
1520 CALL VCHAR(1,16,128,AJ)
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530 ELSE 780
1540 CALL CLEAR :: DISPLAY A
T(1,1):">G GRAPHIK" :: DISPL
AY AT(3,4):"PRO POS.IN %" ::
 DISPLAY AT(5,1):">T TABELLE
"
1550 DISPLAY AT(7,1):">P %AU
SWERTUNG" :: DISPLAY AT(9,1)
:">M MITTELWERT" :: DISPLAY
AT(11,1):">C HARDCOPY" :: DI
SPLAY AT(13,1):">E ENDE"
1560 DISPLAY AT(15,1):">F BE
R.NACH FORMELN"
1570 CALL KEY(0,K,S):: IF S=
0 THEN 1570 ELSE 370
1580 ON ERROR 1600 :: OPEN #
1:DR$,OUTPUT :: FOR Y=1 TO 2
4 :: Q$="" :: FOR X=32 TO 1
STEP -1 :: CALL GCHAR(Y,X,WE
):: WT$=CHR$(WE):: Q$=WT$+Q$
1590 NEXT X :: PRINT #1:TAB(
22);Q$ :: NEXT Y :: CLOSE #1
 :: GOTO 370
1600 DISPLAY AT(5,1)ERASE AL
L:"DATA ODER FILE ERROR !" :
: DISPLAY AT(9,1):"PROGRAMM
ABBRECHEN/NEUSTART!"
1610 CALL KEY(0,K,S):: IF S=
0 THEN 1610 ELSE 370
1620 IF Z1>0 THEN P01=Z1-Z1+
1 :: IF Z2>0 THEN P02=Z2-Z2+
1 :: IF Z3>0 THEN P03=Z3-Z3+
1 :: IF Z4>0 THEN P04=Z4-Z4+
1 :: IF Z5>0 THEN P05=Z5-Z5+
1
1630 IF Z6>0 THEN P06=Z6-Z6+
1 :: IF Z7>0 THEN P07=Z7-Z7+
1 :: IF Z8>0 THEN P08=Z8-Z8+
1 :: IF Z9>0 THEN P09=Z9-Z9+
1 :: IF Z10>0 THEN P010=Z10-
Z10+1
1640 PT=P01+P02+P03+P04+P05+
P06+P07+P08+P09+P010
1650 MW=ST/PT :: DISPLAY AT(
24,3):"~mittelwert~~" :: DIS
PLAY AT(24,16):USING "######
.##":MW
1660 CALL KEY(0,K,S):: IF S=
0 THEN 1660 :: IF K=13 THEN
780 ELSE 1660
1670 FOR OI=13 TO 13 :: CALL
 COLOR(OI,14,2):: CALL CHAR(
128,"FFFFFFFFFFFFFFFF"):: N
EXT OI
1680 CALL CLEAR :: CALL VCHA
R(1,28,128,12):: CALL VCHAR(
1,29,128,12):: CALL VCHAR(1,
30,128,12):: DISPLAY AT(1,26
):"tan"
1690 DISPLAY AT(3,26):"cos"
:: DISPLAY AT(5,26):"log" ::
 DISPLAY AT(7,26):"atn" :: D
ISPLAY AT(9,26):"sin" :: DIS
PLAY AT(11,26):"sqr"
1700 DISPLAY AT(24,1):"auswa
hl~!~~t~c~l~a~s~q~~"
1710 CALL KEY(0,K,S):: IF S=
0 THEN 1710 :: IF K=84 THEN
1730 :: IF K=67 THEN 1770 ::
 IF K=76 THEN 1790
1720 IF K=65 THEN 1810 :: IF
 K=83 THEN 1830 :: IF K=81 T
HEN 1850 ELSE 1710
1730 DISPLAY AT(1,25)SIZE(1)
:"X" :: GOSUB 1870
1740 DISPLAY AT(9,1):TAN(AL)
:: DISPLAY AT(11,1):TAN(BE):
: DISPLAY AT(13,1):"GRAD";TA
N(GA*PI/180)
1750 DISPLAY AT(24,1):"h~cop
```

```
y!c~help!h~endele~"
1760 CALL KEY(0,K,S):: IF S=
0 THEN 1760 :: IF K=67 THEN
1580 :: IF K=72 THEN 1540 ::
 IF K=69 THEN 410 ELSE 1670
1770 DISPLAY AT(3,25)SIZE(1)
:"X" :: GOSUB 1870
1780 DISPLAY AT(9,1):COS(AL)
:: DISPLAY AT(11,1):COS(BE):
: DISPLAY AT(13,1):"GRAD";CO
S(GA*PI/180):: GOTO 1750
1790 DISPLAY AT(5,25)SIZE(1)
:"X" :: DISPLAY AT(3,1)SIZE(
4):"WERT" :: ON WARNING NEXT
 :: ACCEPT AT(3,13)SIZE(10)V
ALIDATE(DIGIT,"."):WRT
1800 DISPLAY AT(5,1)SIZE(12)
:"LOGARITHMUS=" :: DISPLAY A
T(7,1):LOG(WRT):: GOTO 1750
1810 DISPLAY AT(7,25)SIZE(1)
:"X" :: GOSUB 1870
1820 DISPLAY AT(11,1):ATN(AL
):: DISPLAY AT(13,1):ATN(BE)
:: DISPLAY AT(15,1):"GRAD";A
TN(GA*180/PI):: GOTO 1750
1830 DISPLAY AT(9,25)SIZE(1)
:"X" :: GOSUB 1870
1840 DISPLAY AT(11,1):SIN(AL
):: DISPLAY AT(13,1):SIN(BE)
:: DISPLAY AT(15,1):"GRAD";S
IN(GA*PI/180):: GOTO 1750
1850 DISPLAY AT(11,25)SIZE(1
):"X" :: DISPLAY AT(3,1)SIZE
(4):"WERT" :: ON WARNING NEX
T :: ACCEPT AT(3,13)SIZE(10)
VALIDATE(DIGIT,"."):WERT
1860 DISPLAY AT(5,1)SIZE(14)
:"QUADRATWURZEL=" :: DISPLAY
 AT(7,1):SQR(WERT):: GOTO 17
50
1870 DISPLAY AT(3,1)SIZE(11)
:"BOGENMASS 1" :: ON WARNING
 NEXT :: ACCEPT AT(3,13)SIZE
(10)VALIDATE(DIGIT,"."):AL
1880 DISPLAY AT(5,1)SIZE(11)
:"BOGENMASS 2" :: ACCEPT AT(
5,13)SIZE(10)VALIDATE(DIGIT,
"."):BE :: DISPLAY AT(7,1)SI
ZE(8):"GRADMASS"
1890 ACCEPT AT(7,13)SIZE(10)
VALIDATE(DIGIT,"."):GA
1900 RETURN
```

Die nächste TI REVUE erscheint am 28. Oktober

SERVICE

FÜR BUCHHALTER MUSIKER & SAMMLER



SPICKZETTEL

Nachdem das Programm mit RUN, RUN" DSK1. filename" oder RUN" CS1" gestartet worden ist, erscheint das Titelbild. Dazu ertönt die etwas verfremdete Melodie des Liedes „Yellow Submarine" von den Beatles. Mit einer beliebigen Taste geht es weiter. Man wird nun nach dem Namen gefragt und ob der Drukker angeschlossen ist, um das Zeugnis auszudrukken. Jetzt kann man noch wählen, ob man die Spielanleitung lesen will. Nach all diesen Vorbereitungen beginnt nun endlich das eigentliche Spiel.

Obwohl der arme Schüler Willi weiß, daß heute einige Exen (= Stegreifaufgaben) ins Haus stehen, hat er wie üblich nichts gelernt. Als sein Schutzengel steigst Du nun in Willis Körper. Dadurch ist er zwar auch nicht schlauer, aber Du kannst ihm helfen, Spickpunkte einzusammeln, die von Zeit zu Zeit auf einer Schulbank erscheinen. Das versucht der Lehrer natürlich zu verhindern, indem er ihn verfolgt und Sehstrahlen nach ihm wirft. Wird Willi von einem Sehstrahl getroffen, hat ihn der Lehrer beim Spicken erwischt und Du erntest eine Spicksechs. Du bist aber auch nicht ganz wehrlos, Du kannst nämlich mit U-Haken auf den Lehrer schießen. Triffst Du ihn, ist er fünf Minu-

ten bzw. bis zum nächsten gefressenen Spickpunkt bewußtlos. Danach ist er allerdings noch zorniger, d.h. er fährt schneller und schießt öfter. Hast Du 45 (Spiel-) Minuten überstanden oder bist Du vom Lehrer erwischt worden, ist die erste Stunde zu Ende, und der Computer errechnet die Zeugnisnote aus den erreichten Spickpunkten (wenn man nicht ohnehin schon eine Sechs hat). Jetzt folgt die zweite Schulstunde mit einem anderen Fach. Die Lehrer werden von Stunde zu Stunde strenger, was sich in ihrer Geschwindigkeit und der Häufigkeit der Sehstrahlen zeigt. Sind alle sieben Stunden vorbei, wird aus der Summe der abgegebenen Schüsse und der Zahl der Spicksechser die Betragennote errechnet. Anschließend wird das Zeugnis ausgegeben (wenn der Drucker angeschlossen ist, auch schwarz auf weiß). Durch Tastendruck erscheint die Frage, ob man noch einmal spielen will. Betätigt man die Taste ,,J" beginnt das Spiel von vorne, aber ohne Titelbild.

Noch ein paar Hinweise zum Gebrauch des Programms:

1. Gesteuert wird Willi mit dem Joystick 1. Die Uhaken feuert man mit dem Feuerknopf des Joysticks ab.
2. Während des Lehrerwechsels können über die Funktionstasten noch weitere Funktionen betätigt werden:
   mit „CLEAR" kommt man zur Frage, ob man aufhören möchte.
   mit „AID" kommt man zum Zeugnis.
   mit „BACK" beginnt das Spiel von vorne.

Peter Hlauna

Musikkompositionsprogramm

Dieses Programm ist ganz der Musik gewidmet! Es bietet dem Benutzer die Möglichkeit der Eingabe einer Melodie mit bis zu 125 Noten, wobei ihm die Auswahl zwischen Ein-, Zwei- oder Dreiklang angeboten wird. Die eingegebene Melodie kann nach einer eventuellen Korrektur beliebig oft angehört werden. Zusätzlich besteht die Möglichkeit, die Komposition auf Cassette zu speichern oder andere zu laden. Die im Menü angebotene Einleitung (Wahlnummer 1) gibt Einzelinformationen zur Programmhandhabung.

Bei Titelbild und Schlußbild setzt das Drücken einer beliebigen Taste das Programm fort.

Benötigt werden für das Programm nur die Konsole, ein Datenrecorder und Spaß am TI.

P.S. Dieses Programm darf nur mit der Konsole, ohne Xbasic-Modul und ohne angeschlossene Peripherie gefahren werden, da zur Dimensionierung der Felder fast der gesamte Speicherplatz benötigt wird.

Glückliche Besitzer einer 32-K-Erweiterung können das Programm jedoch auch in Extended-Basic fahren.

Alexander Saberschinsky

Steuerberechnung

Das Programm unterscheidet zwischen sozialversicherungs- und nicht sozialversicherungspflichtigen Arbeitnehmern und errechnet nach Eingabe von Bruttojahreseinkommen, Werbungskosten, Sonderausgaben, Zahl der Kinder und etwaiger Baukosten für ein Eigenheim oder eine Eigentumswohnung das zu versteuernde Einkommen, die jährliche Einkommensteuer, die jährliche Kirchensteuer sowie monatliche Arbeitnehmerbeiträge zur Kranken-, Renten- und Arbeitslosenversicherung und das monatliche Nettoeinkommen unter Berücksichtigung einer etwaigen Steuerersparnis nach § 7b EStG. Sollten die genauen Werte für Werbungskosten und Sonderausgaben noch nicht bekannt sein, zeigt das Programm die familienstandabhängigen Pauschalbeträge für diese Werte extra an. Nach Abschluß jeder Berechnung kann zu einer Druckroutine verzweigt werden, die die Werte „schwarz auf weiß" festhält.

Das Programm ist nicht nur für Steuerberater bzw. Vermögensberater interessant, sondern auch für jeden Arbeitnehmer, der über seine Abgaben Bescheid wissen möchte bzw. errechnen möchte, ob etwa eine Erhöhung der Sonderausgaben über z.B. Lebensversicherung o.ä. zu einer Steuerersparnis führt etc.

Bei der Eingabe der Sozialversicherungsquoten am Anfang des Programms ist es wichtig, nur die Arbeitnehmeranteile in % einzugeben.

Thomas Ehlert

Advantage

Bei dem Spiel Advantage werden verschiedene farbige Kugeln in ein Labyrinth von beweglichen Hebeln geworfen, die je nach ihrer Stellung die Kugeln festhalten bzw. freigeben.

Jeder Spieler muß eine vorgegebene Anzahl von Kugeln je Farbe durch das Labyrinth bringen.

Der Spieler, dessen Kugeln alle das Ende des Labyrinth zuerst erreicht haben, ist Sieger.

TI-Fall:

Herzlich willkommen zu TI-Fall, einer mehr oder weniger gelungenen Nachahmung des berühmten Pitfall. Zu Beginn meiner Version von Pitfall werden Sie mit den Rhythmen von La Cucaracha gleich in die Urwaldatmosphäre eingestimmt.

Und nachdem es sich um eine Nachahmung handelt, ist auch hier das Ziel des Spielers, möglichst viele Bilder unbeschadet zu durchlaufen und dabei so viel Punkte zu sammeln, wie nur möglich. Punkte liegen dabei in Form von Goldbarren umher und um diese aufzusammeln, braucht man nur mit seinem Männchen, genannt Cuthbert, darüberzulaufen.

Daß das Ganze natürlich nicht so einfach ist, versteht sich eigentlich von alleine. Lagerfeuer, Baumstämme, Skorpione, Tümpel und gefährliche Vögel bilden den Reiz des Spieles.

Mit ihrem Joystick haben Sie nun die Möglichkeit, ihren Cuthbert nach links oder rechts laufen zu lassen oder ihn mit dem Fire Button Hindernisse überspringen zu lassen.

Leider ist das Listing nicht besonders übersichtlich und daher nicht so einfach einzutippen, jedoch sollte Sie der Spielspaß dafür entschädigen.

Michael Schneller

```
10 ! *********************
11 ! *                   *
12 ! *      TI-FALL      *
13 ! *                   *
14 ! *   Copyright by    *
15 ! * Michael Schneller *
16 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic    *
22 ! *    Joystick (1)   *
23 ! *                   *
26 ! *  Speicherbelegung *
27 ! *     9093 Bytes    *
28 ! *                   *
29 ! *********************
30 !
100 CALL CHAR(34,"FEFDFBF7EF
DFBF7F7FBFDFEFF7FBFDFE")
110 A$=RPT$("F",16):: CALL C
HAR(112,A$,96,A$,97,"0303030
303030303"):: CALL COLOR(9,1
2,2,10,12,2,11,13,11,1,11,1)
120 CALL CHAR(113,""):: RAND
OMIZE :: CALL SCREEN(2):: CA
LL CLEAR :: CALL MAGNIFY(3)
130 CALL CHAR(100,"0C0C001C1
C3838180018141 4E2820203"&RPT
$("0",32),104,"0C0C000C1C1C1
C1C000C0E0A3E28080C"&RPT$("0
",32))
140 CALL CHAR(108,"181800183
C3A3A1800186483000000"&RPT$(
"0",32))
150 CALL CHAR(116,RPT$("0",1
8)&"708D0303038D72"&RPT$("0"
,17)&"C02F1F9FEF0E01")
160 CALL CHAR(120,RPT$("0",1
6)&"30408F9F7F0F0708"&RPT$("
0",19)&"EB1C0C0C0B14E")
170 CALL CHAR(124,RPT$("F",6
4),128,"C0F0FC"&RPT$("F",20)
&"FCF0C000000000C0F0FCFEFEFC
F0C")
180 CALL CHAR(132,"000000000
30F3F7F7F3F0F030000000000030F3
F"&RPT$("F",20)&"3F0F03")
190 CALL CHAR(136,RPT$("0",1
6)&"01071FFEFE1F0701"&RPT$("
0",16)&"80E0F87F7FF8E08")
200 CALL CHAR(48,"00FEC6C6C6
C6C6FE",49,"0018181818181818
",50,"00FE0606FEC0C0FE",51,"
00FE06063E0606FE")
210 CALL CHAR(52,"00D8D8D8FE
181818",53,"00FEC0C0FE0606FE
",54,"00C0C0C0FEC6C6FE",55,"
00FE06063E060606")
220 CALL CHAR(56,"00FEC6C6FE
C6C6FE",57,"00FEC6C6FE060606
")
230 RESTORE 1020 :: FOR I=65
 TO 90 :: READ C$ :: CALL CH
AR(I,C$):: NEXT I
240 CALL CHAR(92,"0000000000
30CD03030100000000000000000
00000CB3C0C08")
250 RESTORE 1160 :: FOR I=1
TO 5 :: READ V(I):: NEXT I
260 CALL CHAR(33,"FF818181FF
818181",140,"0103070F0F0F0F0
703000102040402 01C0E0B0D8D8B
8F8F0E08")
270 DIM SPS(91),LL(91):: HI=
0 :: M(1)=97 :: M(2)=225 ::
DEF SKS=INT(200*RND)+40
280 CALL CHAR(42,"1082105400
FEFEFEFF818181FF818181"):: C
ALL COLOR(2,16,11)
290 RESTORE 1080 :: FOR LI=1
 TO 91 :: READ SPS(LI),L ::
LL(LI)=1000/L :: NEXT LI
300 CALL CHAR(44,"0000010103
0303070D1E3F767E7D3F1F008080
C0C0C0A0F0F8AC7CBECEFEBCF8")
310 CALL VORSPANN(SPS(),LL()
):: MA=4 :: \=81 :: _=-1 ::
CALL CLEAR :: CALL BAUM :: C
ALL HCHAR(24,15,100,MA-1)::
[(1)=100 :: [(3)=104
320 DISPLAY AT(24,19):"HI:";
HI :: GOTO 600
330 Q=1 :: DISPLAY AT(24,19)
:"HI:";HI
340 CALL JOYST(1,X,Y):: _=-_
 :: CALL KEY(1,K,S):: IF Y=4
 THEN 520 ELSE CALL MOTION(#
1,0,X*1.5):: IF K=18 THEN 81
0 ELSE CALL POSITION(#1,Y1,N
)
350 IF N<15 THEN 510 ELSE IF
 N>240 THEN 600 ELSE CALL PA
TTERN(#1,[(_+2)):: CALL GCHA
R(((\+7)/8)+1,((N+7)/8),H)::
 IF H=32 THEN 540 ELSE IF H=
42 THEN 570
360 ON Q GOTO 340,380,420,43
0,490,450,420
370 GOTO 340
380 C=C+1 :: CALL COINC(ALL,
E):: IF E THEN 910 ELSE IF C
=4 THEN 390 ELSE 370
390 IF O=0 THEN CALL SPRITE(
#2,124,5,81,113,#3,128,5,81,
129,#4,132,5,81,97):: C=0 EL
SE CALL DELSPRITE(#3,#2,#4):
: C=0
400 IF O=0 THEN O=1 ELSE O=0
410 GOTO 370
420 CALL COINC(ALL,E):: IF E
 THEN 910 ELSE 340
```

```
430 CALL POSITION(#2,Y2,X2):
: CALL MOTION(#2,0,SGN(N-X2)
*5):: IF SGN(N-X2)=-1 THEN C
ALL PATTERN(#2,116)ELSE CALL
 PATTERN(#2,120)
440 GOTO 420
450 FE=-FE :: ON FE+2 GOTO 4
60,460,470
460 CALL LOCATE(#2,73,80)::
GOTO 480
470 CALL LOCATE(#2,193,1)
480 GOTO 420
490 ZV=ZV+1 :: IF ZV>5 THEN
ZV=0
500 CALL MOTION(#11,V(ZV),-1
0):: GOTO 420
510 CALL MOTION(#1,0,0):: CA
LL LOCATE(#1,\,20):: GOTO 34
0
520 CALL MOTION(#1,0,0):: CA
LL GCHAR(INT((\+7)/8)-2,INT(
(N+7)/8),H):: IF H=33 THEN 5
30 ELSE 360
530 FOR @=153 TO 81 STEP -9
:: CALL LOCATE(#1,@,121):: C
ALL SOUND(-100,@+100,0):: NE
XT @ :: \=81 :: GOTO 360
540 CALL MOTION(#1,0,0):: CA
LL MOTION(#1,30,0):: FOR @=1
 TO 100 :: NEXT @ :: CALL MO
TION(#1,0,0):: CALL LOCATE(#
1,153,N)
550 \=153 :: CALL SOUND(-100
,-8,0,220,30,220,30,330,30):
: GOTO 340
560 GOTO 360
570 CALL MOTION(#1,0,0,#2,0,
0,#11,0,0):: FOR SOU=0 TO 30
 STEP 2 :: CALL SOUND(-100,3
80,SOU,640,SOU):: NEXT SOU
580 FOR @=1 TO 10 :: P=P+100
 :: CALL PU(P):: CALL SOUND(
-50,843,1):: NEXT @
590 CALL HCHAR(((\+7)/8)+1,(
(N+7)/8),113):: CALL PU(P)::
 GOTO 340
600 CALL DELSPRITE(ALL):: FO
R S=0 TO 30 STEP 2 :: CALL S
OUND(-100,262,S,330,S,392,S)
:: NEXT S
610 P=P+100 :: CALL PU(P)::
Q=INT(7*RND)+1
620 CALL HCHAR(10,1,113,128)
:: CALL VCHAR(15,16,32,4)::
CALL HCHAR(19,1,113,128)
630 ON Q GOSUB 650,660,670,7
10,800,750,780
640 CALL SPRITE(#1,100,5,\,2
0):: GOTO 340
650 CALL LOCH(1):: CALL BAR(
2):: RETURN
660 O,C=0 :: CALL BAR(1):: R
ETURN
670 CALL LOCH(1):: G=INT(2*R
ND)+1 :: FOR @=1 TO G :: CAL
L SPRITE(#@+1,136,12,81,M(@)
):: NEXT @
680 D=INT(2*RND)+1 :: FOR @=
1 TO D :: CALL SPRITE(#@+4,1
36,12,153,M(@)):: NEXT @
690 R=INT(7*RND)-12 :: R1=IN
T(5*RND)-13 :: FOR @=1 TO G
:: CALL MOTION(#@+1,0,R):: N
EXT @
700 FOR @=1 TO D :: CALL MOT
ION(#@+4,0,R1):: NEXT @ :: R
ETURN
710 CALL LOCH(1):: @=INT(2*R
ND):: IF @=1 THEN CALL SPRIT
E(#2,116,16,81,SKS,0,-5)ELSE
 CALL SPRITE(#2,116,16,153,S
KS,0,-5)
720 IF @=1 THEN YG=153 ELSE
YG=81
730 U=INT(2*RND)+1 :: FOR @=
1 TO U :: CALL SPRITE(#@+2,1
36,12,YG,M(@)):: NEXT @ :: ]
=INT(5*RND)-13
740 FOR @=1 TO U :: CALL MOT
ION(#@+2,0,]):: NEXT @ :: RE
TURN
750 CALL BAR(2):: CALL HCHAR
(12,11,34):: CALL HCHAR(12,1
2,35):: CALL SPRITE(#2,44,7,
73,80):: CALL LOCH(2)
760 @=INT(2*RND):: IF @=0 TH
EN FE=-1 ELSE FE=1
770 IF RND>.5 THEN CALL SPRI
TE(#3,136,2,153,240,0,INT(6*
RND)-12):: RETURN ELSE RETUR
N
780 CALL SPRITE(#2,124,5,81,
113,#3,132,5,81,97,#4,128,5,
81,129,#10,140,7,65,97,0,9)
790 CALL BAR(1):: RETURN
800 ZV=0 :: CALL SPRITE(#11,
92,16,65,233):: CALL BAR(1):
: RETURN
810 CALL MOTION(#11,0,-7)::
CALL SOUND(50,380,0):: CALL
PATTERN(#1,108)
820 FOR @=-18 TO 18 STEP 6 :
: CALL MOTION(#1,@,X*2.5)::
CALL POSITION(#1,Y1,N):: IF
N<10 THEN 510 ELSE IF N>240
THEN 600
830 CALL COINC(ALL,E):: IF E
 THEN 850
840 NEXT @ :: CALL MOTION(#1
,0,X*2.5):: CALL LOCATE(#1,\
,N):: GOTO 350
```

```
850 CALL MOTION(#1,0,0,#10,0
,0):: CALL COINC(#1,#10,15,E
):: IF E=0 THEN 910
860 CALL MOTION(#1,0,9,#10,0
,9)
870 FOR S=1 TO 90 :: CALL SO
UND(LL(S),SPS(S),0):: CALL K
EY(1,K,Z):: IF K=18 THEN 890
880 NEXT S :: GOTO 870
890 CALL SOUND(-100,392,0)::
 CALL MOTION(#1,0,0):: CALL
POSITION(#1,Y1,N):: CALL LOC
ATE(#1,\,N)
900 GOTO 360
910 CALL MOTION(#1,0,0):: CA
LL DELSPRITE(#2,#3,#4,#5,#6,
#7,#8):: FOR I=1 TO 10 :: CA
LL SOUND(-10,262,0)
920 CALL SOUND(-10,523,0)::
NEXT I :: CALL MOTION(#1,120
,0)
930 CALL POSITION(#1,Y1,N)::
 IF Y1<185 THEN 930
940 CALL DELSPRITE(#1):: FOR
 @=0 TO 30 STEP 2 :: CALL SO
UND(-50,-6,@):: NEXT @
950 MA=MA-1 :: CALL HCHAR(24
,15,32,3):: IF MA=0 THEN 970
 ELSE CALL HCHAR(24,15,100,M
A-1)
960 GOTO 600
970 CALL DELSPRITE(ALL):: DI
SPLAY AT(17,7):"G A M E   O
V E R" :: IF P>HI THEN HI=P
:: DISPLAY AT(24,19):"HI:";H
I
980 CALL VCHAR(15,16,32,4)
990 FOR S=1 TO 91 :: CALL SO
UND(LL(S),SPS(S),0):: CALL K
EY(0,K,Z):: IF Z=0 THEN
1000 ELSE 1010
1000 NEXT S :: GOTO 990
1010 CALL CLEAR :: P=0 :: GO
TO 310
1020 DATA 00FEC6C6C6FEC6C6,0
0FEC6C6F8C6C6FE,00FEC0C0C0C0
C0FE,00F8C6C6C6C6C6F8,00FEC0
C0F8C0C0FE
1030 DATA 00FEC0C0F0C0C0C0,0
0FEC0C0DEC6C6FE,00C6C6C6FEC6
C6C6,0018181818181818,00FE06
0606C6C6FE
1040 DATA 00CCCCCCF0CCCCCC,0
0C0C0C0C0C0C0FE,00FED6D6C6C6
C6C6,00F6F6F6F6CECECE,00FEC6
C6C6C6C6FE
1050 DATA 00FEC6C6FEC0C0C0,0
0CCCCCCCCCEFD,00FEC6C6FEE0D8
C6,00FEC0C0FE0606FE,00FE1818
18181818
1060 DATA 00C6C6C6C6C6C6FE,0
066666666181818,00C6C6C6D6D6
D6FE,00C6C6C638C6C6C6
1070 DATA 00C6C6FE30303030,0
0FE0606380C0FE

1080 DATA 131,8,131,8,131,8,
175,8,30000,4,220,8,30000,8,
131,8,131,8,131,8
1090 DATA 175,8,30000,4,220,
8,30000,2,30000,8,175,4,175,
8,165,8,165,8,147,8,147,8

1100 DATA 131,2,30000,8,131,
8,131,8,131,8,165,8,30000,4,
196,8,30000,8,131,8,131,8,13
1,8
1110 DATA 165,8,30000,4,196,
8,30000,2,30000,8,262,4,294,
8,262,8,233,8,220,8,196,8
1120 DATA 220,3,175,8,175,8,
131,4,131,8,175,8,175,8,220,
8,220,8,262,4,220,8,220,2
1130 DATA 30000,8,262,4,294,
8,262,8,233,8,220,8,262,8,23
3,3,196,8,196,2,30000,8,131,
4
1140 DATA 131,8,165,8,165,8,
196,8,196,8,233,3,196,8,196,
2,30000,8,233,4,233,8,262,8,
233,8,220,8
1150 DATA 196,8,220,3,175,8,
175,2,30000,8,220,3,175,8,17
5,8,175,8
1160 DATA 10,5,0,-5,-10
1170 SUB BAUM
1180 FOR @=3 TO 21 STEP 9 ::
 DISPLAY AT(1,@):"  ppp" ::
DISPLAY AT(2,@):" pppppp"
1190 DISPLAY AT(3,@):"pppppp
pp" :: DISPLAY AT(4,@):" ppp
ppp" :: DISPLAY AT(5,@):"  p
ppp"
1200 DISPLAY AT(6,@):"...''a
" :: DISPLAY AT(7,@):"...'''
" :: DISPLAY AT(8,@):"..'''"
 :: DISPLAY AT(9,@):"''''''''
"

1210 NEXT @ :: CALL HCHAR(10
,1,113,128):: CALL HCHAR(14,
1,112,32):: CALL HCHAR(19,1,
113,128)
1220 CALL HCHAR(23,1,112,32)
:: SUBEND
1230 SUB TI(Z):: FOR @=1 TO
Z :: NEXT @ :: SUBEND
1240 SUB PU(P):: DISPLAY AT(
24,1)SIZE(7):P :: SUBEND
1250 SUB LOCH([])
1260 ON [] GOTO 1270,1280,12
90
1270 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,8,32,2)::
```

```
CALL HCHAR(13,8,32,2)
1280 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,15,32,3)::
 CALL HCHAR(13,15,32,3):: CA
LL VCHAR(15,16,33,4):: CALL
HCHAR(13,16,33)
1290 @=INT(2*RND):: IF @=1 T
HEN CALL HCHAR(12,25,32,2)::
 CALL HCHAR(13,25,32,2)
1300 SUBEND
1310 SUB VORSPANN(SPS(),LL()
)
1320 CALL BAUM :: CALL HCHAR
(19,1,32,160):: CALL VCHAR(1
7,4,96,5)
1330 CALL HCHAR(17,2,96,5)::
 CALL VCHAR(17,8,96,5):: CAL
L HCHAR(19,10,96,2):: CALL V
CHAR(17,13,96,5)
1340 CALL HCHAR(17,14,96,3):
: CALL HCHAR(19,14,96,2):: C
ALL VCHAR(17,18,96,5):: CALL
 HCHAR(17,19,96,2):: CALL VC
HAR(17,21,96,5)
1350 CALL HCHAR(20,19,96,2):
: CALL VCHAR(17,23,96,5):: C
ALL HCHAR(21,24,96,3):: CALL
 VCHAR(17,28,96,5)
1360 CALL HCHAR(21,29,96,3):
: DISPLAY AT(24,1):"<C>1985
BY MICHAEL SCHNELLER"
1370 FOR @=3 TO 8 :: CALL CO
LOR(@,16,1):: NEXT @
1380 FOR S=1 TO 90 :: CALL S
OUND(LL(S),SPS(S),0):: CALL
KEY(0,K,Z):: IF Z=0 THEN 139
0 ELSE SUBEXIT
1390 NEXT S :: GOTO 1380
1400 SUBEND
1410 SUB BAR(\):: ON \ GOTO
1420,1430
1420 @=INT(5*RND):: IF @>2 T
HEN CALL HCHAR(21,20,42)
1430 @=INT(5*RND):: IF @>2 T
HEN CALL HCHAR(12,23,42)
1440 SUBEND
```



```
10 ! *********************
11 ! *                   *
12 ! *    SPICKZETTEL    *
13 ! *                   *
14 ! *   Copyright by    *
15 ! *   Peter Hlawna    *
16 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic    *
22 ! *    Joystick (1)   *
23 ! *                   *
26 ! *  Speicherbelegung *
27 ! *    11576 Bytes    *
28 ! *                   *
29 ! *********************
30 !
400 RANDOMIZE :: Q=1 :: GES=
3 :: SCHWER=.08 :: ZEIT,COUN
T,P,RU,SCH=0 :: FACH$="RELIG
ION"
410 !
420 ! ZEICHENDATAS
430 !
440 DATA 030F1F312E0E6EF1,BE
DD5D0E07020206,C0F0F88C74707
68F,7DB9BB72E24060
450 DATA 030F1F312E0E6EF1,BE
9DDD4E47020600,C0F0F88C74707
68F,7DBBBA70E0404060
460 DATA 0F1F30203F292620,21
200113100C03,F8FC0602FE4A328
2,4202C0E40418E0
470 DATA 0047CFDFFF7F1F0F,07
03,00E2F3FBFFFEF8F0,E0C0
480 DATA 0505050505050505,,4
040404040404040,
490 DATA 0000010204040404,,0
000804020202020,
500 DATA FFFFFFFFFFE1EFE1,EF
E1EFE1EFE1FFFF,FFFFFFFFFF0FE
F0F,EF0FEF0FEF0FFFFF
510 RESTORE :: FOR I=96 TO 1
23 :: READ A$ :: CALL CHAR(I
,A$):: NEXT I
520 CALL CHAR(142,"010204081
0204080FF")
530 CALL COLOR(12,7,7,10,2,1
6,14,1,16,13,13,13,1,16,16)
540 ! BILDSCHIRMAUFBAU
550 CALL CLEAR :: CALL SCREE
N(2):: CALL MAGNIFY(3)
560 DISPLAY AT(1,3):"0 MIN"&
RPT$(CHR$(129),14)&"PKT    0"
570 FOR I=6 TO 21 STEP 5
580 DISPLAY AT(I,1):"xzxz...
.xzxz....xzxz....xzxzy{y{...
.y{y{....y{y{....y{y{lnln...
.lnln....lnln....lnlnmomo...
.momo....momo....momo"
590 NEXT I
```

»

```
600 CALL TUERAUF :: CALL SPR
ITE(#3,104,2,17,256,0,-7)::
CALL GONG :: CALL TUERZU
610 DISPLAY AT(1,15-LEN(FACH
$)/2)SIZE(LEN(FACH$)):FACH$
620 CALL POSITION(#3,Y,X)::
IF X<25 THEN 640 ELSE CALL K
EY(0,K,S):: IF S<1 THEN 620
630 IF K=1 THEN 1520 ELSE IF
 K=2 OR K=15 THEN CALL CLEAR
 :: CALL DELSPRITE(ALL):: CA
LL HCHAR(1,1,47,768):: GOTO
1780 ELSE 620
640 CALL MOTION(#3,0,0):: FO
R I=1 TO 300 :: NEXT I
650 ! SPIELBEGINN
660 CALL COLOR(12,7,8)
670 CALL HCHAR(23,19,32,2)::
 CALL HCHAR(24,19,32,2)
680 CALL SPRITE(#1,96,14,177
,145)
690 WERT=INT(RND*5)+1 :: ON
WERT GOSUB 1900,1920,1940,19
60,1980
700 ORTV=INT(RND*4)*64+25 ::
 ORTH=INT(RND*4)*40+42
710 CALL SPRITE(#4,128,16,OR
TH,ORTV)
720 CALL JOYST(1,XJ,YJ)
730 CALL MOTION(#1,-2*YJ,2*X
J)
740 IF XJ=0 AND YJ=0 THEN 76
0
750 Q=-Q :: CALL PATTERN(#1,
98+2*Q)
760 CALL KEY(1,K,S)
770 IF K=18 THEN 840
780 CALL COINC(#1,#4,10,A)::
 IF A=-1 THEN 1130
790 ZEIT=ZEIT+1 :: IF ZEIT/1
0=INT(ZEIT/10)THEN DISPLAY A
T(1,2)SIZE(2):USING "##":ZEI
T/10-.5 :: IF ZEIT>=450 THEN
 1170
800 COUNT=COUNT+1 :: IF COUN
T<50 THEN 720
810 CALL POSITION(#1,Y1,X1,#
3,Y2,X2):: CALL MOTION(#3,0,
SGN(X1-X2)*GES)
820 IF RND<SCHWER THEN 970 E
LSE 720
830 ! SCHUSS MAENNCHEN
840 CALL POSITION(#1,Y1,X1):
: CALL SPRITE(#2,116,13,Y1,X
1,-100,0):: SCH=SCH+1 :: P=M
AX(P-1,0)
850 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
860 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
870 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
880 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
890 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
900 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
910 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
920 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940
930 CALL COINC(#3,#2,10,A)::
 IF A=-1 THEN 940 ELSE DISPL
AY AT(1,26):USING "###":P ::
 CALL DELSPRITE(#2):: GOTO 7
80
940 CALL DELSPRITE(#2):: DIS
PLAY AT(1,26):USING "###":P
950 CALL MOTION(#3,0,0):: FO
R I=0 TO 20 STEP 2 :: CALL S
OUND(-100,-5,I):: NEXT I ::
COUNT=0 :: GES=GES+1 :: SCHW
ER=SCHWER+.02 :: GOTO 780
960 ! SCHUSS LEHRER
970 CALL POSITION(#3,Y2,X2):
: CALL SPRITE(#2,112,11,Y2,X
2,100,0)
980 CALL COINC(#1,#2,10,A)::
 IF A=-1 THEN 1090
990 CALL COINC(#1,#2,10,A)::
 IF A=-1 THEN 1090
1000 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1010 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1020 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1030 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1040 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1050 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1060 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1070 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090
1080 CALL COINC(#1,#2,10,A):
: IF A=-1 THEN 1090 ELSE CAL
L DELSPRITE(#2):: GOTO 720
1090 CALL DELSPRITE(#2):: CA
LL MOTION(#1,0,0,#3,0,0)
1100 FOR I=550 TO 110 STEP -
10 :: CALL SOUND(-100,I,5)::
 NEXT I
1110 P=0 :: SCH=SCH+5 :: GOT
O 1170
1120 ! SPICKPUNKTE GEFRESSEN
1130 CALL MOTION(#1,0,0):: C
ALL DELSPRITE(#4):: CALL SOU
```

```
ND(-100,110,5):: CALL SOUND(
-100,330,5):: CALL SOUND(-10
0,500,5)
1140 P=P+WERT :: DISPLAY AT(
1,26):USING "###":P
1150 COUNT=50 :: GOTO 690
1160 ! STUNDE VORBEI
1170 NOTE(RU)=1
1180 IF P<100 THEN NOTE(RU)=
2
1190 IF P<80 THEN NOTE(RU)=3
1200 IF P<60 THEN NOTE(RU)=4
1210 IF P<40 THEN NOTE(RU)=5
1220 IF P<20 THEN NOTE(RU)=6
1230 CALL DELSPRITE(#4):: CA
LL MOTION(#1,0,0,#3,0,0)
1240 CALL GONG :: CALL COLOR
(12,7,7):: CALL TUERAUF :: C
ALL MOTION(#3,0,7)
1250 CALL POSITION(#3,Y2,X2)
:: IF X2>250 THEN 1280 ELSE
CALL KEY(0,K,S):: IF S=0 THE
N 1250
1260 IF K=1 THEN 1520 ELSE I
F K=2 OR K=15 THEN CALL CLEA
R :: CALL DELSPRITE(ALL):: C
ALL HCHAR(1,1,47,768):: GOTO
 1780
1270 GOTO 1250
1280 CALL DELSPRITE(#3):: CA
LL TUERZU :: CALL HCHAR(1,10
,129,14)
1290 RU=RU+1 :: ON RU GOSUB
1340,1370,1400,1430,1460,149
0,1510
1300 ZEIT,COUNT,P=0 :: CALL
HCHAR(1,27,32,5):: CALL HCHA
R(1,1,32,5)
1310 GOTO 600
1320 ! NEUE STUNDE
1330 ! ERDKUNDE
1340 CALL CHAR(104,"0F1F1804
0F1424380E0F031F3F7F7C30F038
7C7EFE666F57070686C8F")
1350 FACH$="ERDKUNDE" :: GES
=4 :: SCHWER=.08 :: RETURN
1360 ! BIOLOGIE
1370 CALL CHAR(104,"0730405C
422869616240422F3C1B0F07E018
047484282C0C8C0484E878B0E0C0
")
1380 SCHWER=.08 :: FACH$="BI
OLOGIE" :: GES=4 :: RETURN
1390 ! LATEIN
1400 CALL CHAR(104,"00070E10
08101F30508870100C13080700F0
183C3C66D62A0A2A1204040830C0
")
1410 SCHWER=.12 :: FACH$="LA
TEIN" :: GES=5 :: RETURN
1420 ! ENGLISCH
1430 CALL CHAR(104,"030F3C20
1010213F60A0C422211D101F80E0
7070F8F8FCBC1C5C3C1818284080
")
1440 SCHWER=.13 :: FACH$="EN
GLISCH" :: GES=5 :: RETURN
1450 ! DEUTSCH
1460 CALL CHAR(104,"071E3832
30101008080908090A040201F03C
0E2606048488884808C8281020C0
")
1470 SCHWER=.15 :: FACH$="DE
UTSCH" :: GES=6 :: RETURN
1480 ! MATHEMATIK
1490 CALL CHAR(104,"0A0F1F38
3235121010111010130908O7A8F8
FC0E2656240404440404E4C808F0
")
1500 SCHWER=.17 :: FACH$="MA
THEMATIK" :: GES=6 :: RETURN
1510 ! ZEUGNIS
1520 CALL CLEAR :: CALL DELS
PRITE(ALL):: CALL CHARSET ::
 CALL HCHAR(1,1,47,768):: CA
LL TAFEL
1530 CALL SCHRIFT(6,9,"ABSCH
LUSSZEUGNIS")
1540 CALL SCHRIFT(8,9,"FUER
"&NAME$)
1550 CALL SCHRIFT(10,9,"RELI
GION......"&STR$(NOTE(0))):: 
 CALL SCHRIFT(11,9,"ERDKUNDE
......"&STR$(NOTE(1)))
1560 CALL SCHRIFT(12,9,"BIOL
OGIE......"&STR$(NOTE(2))):: 
 CALL SCHRIFT(13,9,"LATEIN..
......"&STR$(NOTE(3)))
1570 CALL SCHRIFT(14,9,"ENGL
ISCH......"&STR$(NOTE(4))):: 
 CALL SCHRIFT(15,9,"DEUTSCH.
......"&STR$(NOTE(5)))
1580 CALL SCHRIFT(16,9,"MATH
EMATIK...."&STR$(NOTE(6)))
1590 IF SCH<26 THEN BETR$="S
EHR GUT"
1600 IF SCH>25 THEN BETR$="G
UT"
1610 IF SCH>50 THEN BETR$="S
CHLECHT"
1620 IF SCH>75 THEN BETR$="S
EHR SCHLECHT"
1630 CALL SCHRIFT(18,9,"BETR
AGEN: "&BETR$)
1640 IF D$<>"J" AND D$<>"j"
THEN 1750
1650 ! ZEUGNIS AUSDRUCKEN
1660 OPEN #1:"RS232"
1670 PRINT #1:TAB(6);"ABSCHL
USSZEUGNIS": : : : :
1680 PRINT #1:TAB(4);"fuer "
```

```
;NAME$: :"Schueler des TI-Gy
mnasiums": : :
1690 PRINT #1:TAB(6);"Religi
on........";STR$(NOTE(0)): :
TAB(6);"Erdkunde........";ST
R$(NOTE(1)):
1700 PRINT #1:TAB(6);"Biolog
ie........";STR$(NOTE(2)): :
TAB(6);"Latein..........";ST
R$(NOTE(3)):
1710 PRINT #1:TAB(6);"Englis
ch........";STR$(NOTE(4)): :
TAB(6);"Deutsch.........";ST
R$(NOTE(5)):
1720 PRINT #1:TAB(6);"Mathem
atik......";STR$(NOTE(6)): :
 : :
1730 PRINT #1:"Das Betragen
des Schuelers war durchwegs
";BETR$;".":"Die Mitarbeit l
iess zu wuenschen uebrig."
1740 CLOSE #1
1750 FOR I=1 TO 500 :: CALL
KEY(0,K,S):: IF S<>0 THEN 17
90
1760 NEXT I
1770 ! NEUES SPIEL ?
1780 IF K=15 THEN 1820
1790 CALL TAFEL :: CALL SCHR
IFT(8,3,"WILLST DU WEITERMAC
HEN ODER.....GLEICH VON DER
SCHULE GEHEN?")
1800 CALL KEY(0,K,S):: IF S=
0 THEN 1800
1810 IF K<>74 AND K<>106 THE
N END
1820 CALL TAFEL
1830 CALL SCHRIFT(10,14,"NAM
E:")
1840 ACCEPT AT(13,8)VALIDATE
(UALPHA)SIZE(-14):NAME$
1850 CALL TAFEL :: CALL SCHR
IFT(10,3,"IST DER DRUCKER AN
GESCHLOSSEN")
1860 CALL KEY(0,K,S):: IF S=
0 THEN 1860
1870 D$=CHR$(K):: FOR I=0 TO
 6 :: NOTE(I)=0 :: NEXT I
1880 CALL TAFEL :: GOTO 400
1890 ! CHARDEFINITIONEN FUER
 SPICKPUNKTE
1900 CALL CHAR(128,"00000000
03070F0303030303",130,"",131
,"")
1910 RETURN
1920 CALL CHAR(128,"00000000
0307060000010307070000000000
0000C0E06060E0C080E0E0")
1930 RETURN
1940 CALL CHAR(128,"00000000
0307000303000703000000000000
0000C0E060E0E060E0C0")
1950 RETURN
1960 CALL CHAR(128,"00000000
0606060607070000000000000000
00000000C0C0E0E0C0C0")
1970 RETURN
1980 CALL CHAR(128,"00000000
0707060707000707000000000000
0000E0E000E0E060E0E0")
1990 RETURN
2000 DATA 500,784,500,659,50
0,587,500,659,300,44733
2010 DATA 500,659,500,784,50
0,784,200,698,200,659,500,58
7,300,44733
2020 DATA 400,659,400,622,40
0,659,400,622,400,659,400,62
2,500,784
2030 DATA 200,330,200,311,20
0,330,200,311,200,330,200,31
1,300,349
2040 DATA 200,262,300,262,40
0,294
2050 ! UNTERPROGRAMME
2060 SUB TAFEL
2070 CALL CHAR(47,"FFFFFFFFF
FFFFFFF",128,"FEFEFEFEFEFEFE
FE")
2080 CALL COLOR(13,13,2,9,15
,15,1,13,13,0,13,13):: FOR I
=2 TO 8 :: CALL COLOR(I,16,1
3):: NEXT I
2090 FOR I=19 TO 24 :: CALL
HCHAR(I,11,96,12):: NEXT I
2100 FOR I=1 TO 7 :: CALL VC
HAR(6,I,32,13):: NEXT I
2110 CALL VCHAR(6,8,128,13)
2120 FOR I=9 TO 23 :: CALL V
CHAR(6,I,32,13):: NEXT I
2130 CALL VCHAR(6,24,128,13)
2140 FOR I=25 TO 32 :: CALL
VCHAR(6,I,32,13):: NEXT I
2150 SUBEND
2160 SUB TITEL(D$,NAME$)
2170 CALL CLEAR :: CALL SCRE
EN(2):: CALL HCHAR(1,1,47,76
8):: CALL TAFEL
2180 CALL SCHRIFT(8,11,"WILL
KOMMEN ZU"):: CALL VCHAR(12,
12,47,5):: CALL HCHAR(12,13,
47,3)
2190 CALL HCHAR(16,13,47,3):
: CALL HCHAR(14,13,47)
2200 FOR I=17 TO 21 :: CALL
HCHAR(I-5,I,47):: CALL HCHAR
(33-I,I,47):: NEXT I
2210 CALL MUSIK :: CALL TAFE
L
2220 CALL SCHRIFT(10,14,"NAM
E:"):: ACCEPT AT(13,8)VALIDA
```

```
TE(UALPHA)SIZE(-14):NAME$
2230 CALL TAFEL
2240 CALL SCHRIFT(10,3,"IST
DER DRUCKER ANGESCHLOSSEN")
2250 CALL KEY(0,K,S):: IF S=
0 THEN 2250
2260 D$=CHR$(K)
2270 CALL TAFEL :: CALL SCHR
IFT(10,3,"WILLST DU DIE SPIE
LANLEITUNG")
2280 CALL KEY(0,K,S):: IF S=
0 THEN 2280
2290 IF K<>74 AND K<>106 THE
N CALL TAFEL :: GOTO 2320
2300 CALL TAFEL :: CALL SCHR
IFT(7,2,"OBWOHL DU WEISST,DA
SS IHR       HEUTE EINIGE EX
EN SCHREIBEN     WERDET,HAST
 DU NICHTS GELERNT.")
2310 CALL SCHRIFT(13,2,"SAMM
LE SPICKPUNKTE UND LEGE.....
LEHRER MIT UHAKEN LAHM. ....
....HUETE DICH VOR SEHSTRAHL
EN.")
2320 SUBEND
2330 SUB GONG
2340 FOR I=550 TO 220 STEP -
110 :: FOR J=0 TO 18 :: CALL
 SOUND(-100,I,J):: NEXT J ::
 NEXT I
2350 SUBEND
2360 SUB TUERAUF
2370 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,142):: NEXT I
2380 FOR I=1 TO 100 :: NEXT
I
2390 FOR I=1 TO 3 :: CALL HC
HAR(5-I,30,32,3):: NEXT I
2400 CALL HCHAR(2,29,143,4)
2410 SUBEND
2420 SUB TUERZU
2430 CALL HCHAR(2,29,32,4)
2440 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,142):: NEXT I
2450 FOR I=1 TO 100 :: NEXT
I
2460 FOR I=1 TO 3 :: CALL HC
HAR(5-I,29+I,32):: NEXT I
2470 SUBEND
2480 SUB SCHRIFT(Z,S,A$)
2490 FOR I=1 TO LEN(A$)
2500 IF S-1+I>32 THEN Z=Z+2
:: S=S-32
2510 CALL HCHAR(Z,S-1+I,ASC(
SEG$(A$,I,1)))
2520 FOR J=1 TO 10 :: NEXT J
 :: NEXT I
2530 SUBEND
2540 SUB MUSIK
2550 FOR J=1 TO 2 :: RESTORE
 2610 :: FOR I=1 TO 23 :: RE
AD A,B,C,D :: CALL SOUND(A,B
,0,C,9,D,9)
2560 CALL KEY(0,K,S):: IF S<
>0 THEN 2700
2570 NEXT I :: NEXT J
2580 FOR J=1 TO 2 :: RESTORE
 2660 :: FOR I=1 TO 20 :: RE
AD A,B,C,D :: CALL SOUND(A,B
,0,C,9,D,9)
2590 CALL KEY(0,K,S):: IF S<
>0 THEN 2700
2600 NEXT I :: NEXT J :: GOT
O 2550
2610 DATA 300,494,494,494,10
0,523,523,523,400,587,147,14
7,400,587,185,220,300,587,18
5,220,100,494,185,220
2620 DATA 300,440,262,165,10
0,494,262,165,400,392,196,19
6,400,392,294,247,400,392,29
4,247
2630 DATA 300,494,494,494,10
0,494,494,494,300,440,220,22
0,100,392,220,220,400,330,16
5,165,300,330,262,220
2640 DATA 100,330,262,220,30
0,494,196,196,100,494,196,19
6,400,440,185,185,400,440,22
0,220,800,440,262,262
2650 ! DATA 400,44733,44733,
44733
2660 DATA 400,587,247,196,40
0,587,247,196,400,587,247,19
6,300,587,247,196,100,659,24
7,196
2670 DATA 300,440,220,185,10
0,440,220,185,300,440,220,18
5,100,440,220,185,800,440,22
0,185
2680 DATA 300,440,220,185,10
0,440,220,185,300,440,220,18
5,100,440,220,185,800,440,22
0,185
2690 DATA 300,392,247,196,10
0,392,247,196,300,392,247,19
6,100,392,247,196,800,392,24
7,196
2700 SUBEND
2710 !
2720 ! ********************
2730 !
2740 !  AEHNLICHKEITEN DER
2750 !  LEHRER MIT LEBEN-
2760 !  DEN ODER TOTEN PER-
2770 !  SONEN SIND REIN ZU-
2780 !      FAELLIG !!
2790 !
2800 ! ********************
```

```
10 ! *********************
11 ! *                   *
12 ! *     ADVANTAGE     *
13 ! *                   *
14 ! *   Copyright by    *
15 ! *  Joerg Ruebesamen *
16 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic    *
22 ! *                   *
26 ! *  Speicherbelegung *
27 ! *     4025 Bytes    *
28 ! *                   *
29 ! *********************
120 CALL CLEAR
130 FOR I=1 TO 72 :: PRINT "
ADVANTAGE";
140 NEXT I
150 DISPLAY AT(10,1):".......
............................R
UEBIOLA 1984 LTD...........
......................."
160 OPTION BASE 1
170 FOR I=1 TO 7 :: CALL COL
OR(I,16,15):: NEXT I :: CALL
 COLOR(10,10,15,11,16,15,12,
2,15,9,13,15,8,16,15,0,15,15
)
180 DIM AM(3),K(4,30,2),C(3)
190 C(1)=120 :: C(2)=104 ::
C(3)=112
200 IF RND>.5 THEN PL=1 ELSE
 PL=1.5
210 RANDOMIZE
220 ON WARNING NEXT
230 DATA C0FFFFFF1C0E0703,03
01,03FFFFFF3870E0C0,C0E0F078
3C1E0F07,0000000000000008
240 DATA 0000000000000001,03
070F1E3C78F0E,C08,1818181818
181818
250 DATA 4,4,4,10,4,16
260 DATA 9,1,9,7,9,13,9,19
270 DATA 14,4,14,10,14,16
280 DATA 19,1,19,7,19,13,19,
19
290 A$="0000000A041F040A" ::
 CALL CHAR(120,A$,104,"00000
01F1F1F1F1F",112,A$)
300 FOR I=88 TO 96 :: READ A
$ :: CALL CHAR(I,A$):: NEXT
I
310 CALL HCHAR(1,1,30,768)
320 FOR I=6 TO 21 STEP 3 ::
CALL HCHAR(1,I,49+A):: A=A+1
 :: NEXT I
330 FOR I=1 TO 27 STEP 3 ::
CALL VCHAR(1,I,96,24):: NEXT
 I
340 FOR I=1 TO 14 :: READ A,
B :: IF RND<.5 THEN CALL SPL
(A,B)ELSE CALL SPR(A,B)
350 NEXT I
360 RESTORE 250 :: FOR A=1 T
O 14 :: READ I,J
370 DISPLAY AT(I-1,J+1)SIZE(
2):CHR$(30)&CHR$(30):: NEXT
A
380 I=2 :: J=9
390 PL=PL+.5 :: DISPLAY AT(2
2,24):">>"&STR$(INT(INT(PL)/
PL)+1)&"<<"
400 DISPLAY AT(1,24):"SP.:1"
 :: DISPLAY AT(11,24):"SP.:2
"
410 B=4 :: CALL PUNKT(B,A,PL
,K(,,),C())
420 DISPLAY AT(24,1):" x=1
 h=2   p=3  " :: ACCEPT AT(2
4,20)SIZE(1)VALIDATE("123")B
EEP:F :: DISPLAY AT(24,1):"W
ELCHE SPALTE ?"
430 ACCEPT AT(24,20)SIZE(1)V
ALIDATE("123456")BEEP:SP
440 J=SP*3+2 :: I=2
450 FOR A=1 TO 30 :: IF K(F,
A,1)=0 THEN 470
460 NEXT A
470 IF A>AM(F)THEN AM(F)=A
480 K(F,A,1)=I :: K(F,A,2)=J
 :: CALL HCHAR(K(F,A,1),K(F,
A,2),C(F))
490 CALL ROLL(K(,,),C(),F,A,
PL):: IF @=0 THEN 500 ELSE 4
90
500 CALL SOUND(10,1000,10)::
 FOR T=1 TO 3 :: FOR B=1 TO
3
510 FOR A=1 TO AM(B):: IF K(
B,A,2)>0 THEN CALL ROLL(K(,,
),C(),B,A,PL)
520 NEXT A
530 NEXT B
540 NEXT T
550 GOTO 390
560 SUB SPL(I,J)
570 CALL SOUND(100,-6,20)
580 DISPLAY AT(I,J)SIZE(3):"
XZ"&CHR$(30):: DISPLAY AT(I+
1,J)SIZE(3):"Y[\" :: DISPLAY
 AT(I+2,J)SIZE(3):CHR$(30)&"
Y["
590 SUBEND
600 SUB SPR(I,J)
610 CALL SOUND(100,-5,20)
620 DISPLAY AT(I,J)SIZE(3):C
HR$(30)&"XZ" :: DISPLAY AT(I
+1,J)SIZE(4):"]^_"&CHR$(30):
: DISPLAY AT(I+2,J)SIZE(4):"
^_"&CHR$(30)&CHR$(30)
```

```
630 I=I+2
640 SUBEND
650 SUB ROLL(K(,,),C(),B,A,P
L)
660 I=K(B,A,1):: J=K(B,A,2):
: GOTO 680
670 K(B,A,1)=I :: K(B,A,2)=J
680 CALL GCHAR(I+1,J,G):: CA
LL GCHAR(I+1,J+1,G1)
690 IF G<>120 AND G<>104 AND
 G<>112 THEN 720 ELSE IF I<>
2 AND I<>7 AND I<>12 AND I<>
17 THEN 780
700 CALL GCHAR(I+1,J+2,G)::
CALL GCHAR(I+1,J-1,G1):: IF
G1<>30 AND G<>30 THEN 780 EL
SE CALL HCHAR(I,J,30):: I=I+
2 :: IF G1=96 THEN J=J+3 ELS
E IF G=96 THEN J=J-3
710 CALL HCHAR(I,J,C(B)):: C
ALL SOUND(20,-2,0):: GOTO 67
0
720 IF G+G1<>60 THEN 750
730 CALL HCHAR(I,J,30):: I=I
+1 :: IF I>22 THEN CALL VCHA
R(23,K(B,A,2),30,1):: K(B,A,
1),K(B,A,2)=0 :: CALL PUNKT(
B,A,PL,K(,,),C()):: GOTO 780
 ELSE CALL HCHAR(I,J,C(B))
740 GOTO 680
750 IF G>30 AND G<91 OR G1>3
0 AND G1<91 THEN 770
760 IF G1>30 THEN CALL SPL(I
,J-1):: GOTO 730 ELSE IF G>3
0 THEN CALL SPR(I,J-4):: GOT
O 730
770 K(B,A,1)=I :: K(B,A,2)=J
 :: @=0
780 SUBEND
790 SUB PUNKT(B,A,PL,K(,,),C
())
800 DIM FI(4,4,2),FE(4,4,2)
810 IF BRETT=0 THEN CALL BR(
FE(,,),FI(,,),C()):: BRETT=8
820 K(B,A,2),K(B,A,1)=0
830 IF PL=INT(PL)THEN 890
840 SUM,Z=0
850 FOR I=1 TO 4 :: FOR J=1
TO 4 :: SUM=SUM+FE(I,J,2)::
IF FE(I,J,1)=B AND FE(I,J,2)
=1 AND Z=0 THEN CALL HCHAR(I
+2,J+26,30):: FE(I,J,2)=7 ::
 Z=2 :: SUM=SUM+6
860 NEXT J :: NEXT I
870 IF SUM>=16*7 THEN CALL E
NDE(1)
880 SUBEXIT
890 SUM,Z=0
900 FOR I=1 TO 4 :: FOR J=1
TO 4 :: SUM=SUM+FI(I,J,2)::
IF FI(I,J,1)=B AND FI(I,J,2)
=1 AND Z=0 THEN CALL HCHAR(I
+12,J+26,30):: FI(I,J,2)=7 :
: Z=2 :: SUM=SUM+6
910 NEXT J :: NEXT I
920 IF SUM=16*7 THEN CALL EN
DE(2)
930 SUBEND
940 SUB BR(FE(,,),FI(,,),C()
)
950 FOR I=1 TO 4 :: FOR J=1
TO 4
960 IF RND<.4 THEN FI(I,J,1)
=1 ELSE IF RND<.65 THEN FI(I
,J,1)=2 ELSE FI(I,J,1)=3
970 CALL HCHAR(I+12,J+26,C(F
I(I,J,1)))
980 IF RND<.4 THEN FE(I,J,1)
=1 ELSE IF RND<.65 THEN FE(I
,J,1)=2 ELSE FE(I,J,1)=3
990 CALL HCHAR(I+2,J+26,C(FE
(I,J,1)))
1000 FE(I,J,2),FI(I,J,2)=1
1010 NEXT J :: NEXT I
1020 SUBEND
1030 SUB ENDE(I)
1040 DISPLAY AT(10,1)BEEP ER
ASE ALL:"SPIELER";I;"HAT GEW
ONNEN"
1050 DISPLAY AT(14,1):"1..NO
CHMAL.......................
........................2..SC
HLUSS"
1060 CALL KEY(0,K,S):: IF K=
49 THEN RUN ELSE IF K=50 THE
N STOP ELSE CALL SOUND(4201,
262,16,370,17,311,17):: GOTO
 1060
1070 SUBEND
```

BÖRSE

```
10 ! **********************
11 ! *                    *
12 ! *  STEUERBERECHNUNG  *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *   Thomas Ehlert    *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *   opt.: Drucker    *
23 ! *                    *
26 ! * Speicherbelegung   *
27 ! *     ?70? Bytes     *
28 ! *                    *
29 ! **********************
30 !
280 CALL CLEAR
290 CALL SCREEN(12)
300 DISPLAY AT(1,6):"STEUERB
ERECHNUNG"
310 DISPLAY AT(2,6):"-------
---------"
320 DISPLAY AT(3,6):"@ T.Ehl
ert  3/85 "
330 DISPLAY AT(5,1):"DAS PRO
GRAMM BERECHNET JAH-"
340 DISPLAY AT(6,1):"RESSTEU
ER,KIRCHENSTEUER UND"
350 DISPLAY AT(7,1):"MONATLI
CHES NETTOEINKOMMEN"
360 DISPLAY AT(8,1):"UNTER B
ERUECKSICHTIGUNG EI-"
370 DISPLAY AT(9,1):"NER ETW
AIGEN STEUERERSPARNIS"
380 DISPLAY AT(10,1):"NACH P
ARAGRAPH 7b EStG, SO-"
390 DISPLAY AT(11,1):"WIE SA
EMTLICHE SOZIALVERSI-"
400 DISPLAY AT(12,1):"CHERUN
GSBEITRAEGE VON SOZI-"
410 DISPLAY AT(13,1):"ALVERS
ICHERUNGSPFLICHTIGEN"
420 DISPLAY AT(14,1):"ARBEIT
NEHMERN"
430 DISPLAY AT(20,7):"EINE T
ASTE DRUECKEN!"
440 CALL KEY(0,K,S):: IF S=0
 THEN 440
450 CALL CLEAR
460 DISPLAY AT(1,1):""
470 DISPLAY AT(2,1):"ERKLAER
UNG DER EINGABEWERTE:"
480 DISPLAY AT(3,1):"-------
---------------------"
490 DISPLAY AT(5,1):"~BRUTTO
JAHRESEINKOMMEN BEIN-HALTET
SAEMTLICHE STEUER-...PFLICHT
IGEN EINKUENFTE, ALSO AUCH U
RLAUBS- UND WEIHNACHTSGELD"
500 DISPLAY AT(9,1):"SOWIE S
TEUERPFLICHTIGE SPAR-ZULAGEN
 ETC. ...............~WERBUN
GSKOSTEN SIND IN TAT-SAECHLI
CHER HOEHE, HILFSWEI-"
510 DISPLAY AT(13,1):"SE ALS
 PAUSCHALBETRAG EINZU-GEBEN.
"
520 DISPLAY AT(15,1):"~SONDE
RAUSGABEN SIND IN TAT-SAECHL
ICHER HOEHE, HILFSWEI-"
530 DISPLAY AT(17,1):"SE ALS
 PAUSCHALBETRAG EINZU-GEBEN.
"
540 DISPLAY AT(19,1):"~HAUS-
BAUKOSTEN BEINHALTEN  NUR DI
E HERSTELLUNGS- NICHT ABER D
IE GRUNDSTUECKSKOSTEN"
550 DISPLAY AT(22,7):"EINE T
ASTE DRUECKEN!"
560 CALL KEY(0,K,S):: IF S=0
 THEN 560
570 CALL CLEAR
580 DISPLAY AT(2,1):"IM FOLG
ENDEN ERHALTEN SIE...DIE ZUR
 HILFSWEISEN EINGABE ERFORDE
RLICHEN"
590 DISPLAY AT(5,6):"PAUSCHA
LBETRAEGE"
600 DISPLAY AT(7,6):"-------
---------"
610 DISPLAY AT(9,1):"FUER WE
RBUNGSKOSTEN UND SON-DERAUSG
ABEN:"
620 DISPLAY AT(12,1):"ABKUER
ZUNGSERKLAERUNG:"
630 DISPLAY AT(13,1):"LED=LE
DIG VH=VERHEIRATET....SVP=SO
ZIALVERS.PFLICHTIG....DOP=DO
PPELVERDIENER"
640 DISPLAY AT(17,1):"SIND S
IE (1) LED + SVP............
...(2) LED + NICHT SVP......
...(3) VH  + SVP............
...(4) VH  + NICHT SVP......
...(5) DOP + SVP"
650 DISPLAY AT(22,1):"......
...(6) DOP + NICHT SVP......
........................BITTE
KENNZAHL EINGEBEN!"
660 ACCEPT AT(24,26)BEEP VAL
IDATE("123456")SIZE(1):Q
670 IF Q=1 THEN 730
680 IF Q=2 THEN 750
690 IF Q=3 THEN 770
700 IF Q=4 THEN 790
710 IF Q=5 THEN 810
720 IF Q=6 THEN 830
730 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
```

```
LE  564"
740 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE 3780" :: GO
TO 850
750 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
760 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE 2268" :: GO
TO 850
770 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
780 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....7560.........EIN  K
IND......8424.........ZWEI K
INDER....8656.........DREI K
INDER....9130" :: GOTO 850
790 CALL CLEAR :: DISPLAY AT
(5,1):"WERBUNGSKOSTENPAUSCHA
LE  564"
800 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....4536.........EIN  K
IND......5400.........ZWEI K
INDER....6318.........DREI K
INDER....7236" :: GOTO 850
810 CALL CLEAR :: FB=2 :: DI
SPLAY AT(5,1):"WERBUNGSKOSTE
NPAUSCHALE 1128"
820 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....7560.........EIN  K
IND......8424.........ZWEI K
INDER....9342.........DREI K
INDER...10260" :: GOTO 850
830 CALL CLEAR :: FB=2 :: DI
SPLAY AT(5,1):"WERBUNGSKOSTE
NPAUSCHALE 1128"
840 DISPLAY AT(8,1):"SONDERA
USGABENPAUSCHALE......OHNE K
INDER....4536.........EIN  K
IND......5400.........ZWEI K
INDER....6318.........DREI K
INDER....7236" :: GOTO 850
850 DISPLAY AT(18,1):"BITTE
MERKEN SIE SICH DIE...FUER S
IE RELEVANTEN BETRAEGEUND ST
ARTEN SIE DAS PROGRAMMMIT EI
NEM DEZENTEN DRUCK AUFIRGEND
 EINE TASTE!"
860 CALL KEY(0,K,S):: IF S=0
 THEN 860
870 CALL CLEAR
880 FB=1
890 DISPLAY AT(5,6):"BITTE G
EBEN SIE DIE"
900 DISPLAY AT(7,6):"ARBEITN
EHMERANTEILE"
910 DISPLAY AT(9,6):"IN % FU
ER FOLGENDE"
920 DISPLAY AT(11,6):"VERSIC
HERUNGEN EIN:"
930 DISPLAY AT(17,1):"KRANKE
NVERSICHERUNG?"
940 ACCEPT AT(17,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):KV
950 DISPLAY AT(19,1):"RENTEN
VERSICHERUNG ?"
960 ACCEPT AT(19,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):RV
970 DISPLAY AT(21,1):"ARBEIT
SLOSENVERS.  ?"
980 ACCEPT AT(21,22)BEEP VAL
IDATE(NUMERIC)SIZE(5):ALV
990 CALL CLEAR
1000 DISPLAY AT(2,1):"BRUTTO
JAHRESEINK.? "
1010 ACCEPT AT(2,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):BJ
1020 DISPLAY AT(3,1):"WERBUN
GSKOSTEN...? "
1030 ACCEPT AT(3,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):WK
1040 DISPLAY AT(4,1):"SONDER
AUSGABEN...? "
1050 ACCEPT AT(4,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):SA
1060 DISPLAY AT(5,1):"HAUS-B
AUKOSTEN...? "
1070 ACCEPT AT(5,21)BEEP VAL
IDATE(NUMERIC)SIZE(9):HBK
1080 DISPLAY AT(6,1):"ZAHL D
ER KINDER  ? "
1090 ACCEPT AT(6,21)BEEP VAL
IDATE(NUMERIC)SIZE(2):KZ
1100 DISPLAY AT(7,1):"SPLITT
ING(J=2/N=1) "
1110 ACCEPT AT(7,21)BEEP VAL
IDATE(NUMERIC)SIZE(1):SP
1120 IF HBK>250000 THEN HBK=
250000
1130 STSP=(HBK/100)*5
1140 JG=BJ-SA-WK-(1080*FB)-(
KZ*432)-STSP
1150 A=JG
1160 ON SP GOTO 1170,1190
1170 V=INT(JG/54)*54
1180 GOTO 1200
1190 V=(INT(JG/108)*108)/SP
1200 IF V<4213 THEN 1250
1210 IF V<18000 THEN 1260
1220 IF V<60000 THEN 1280
1230 IF V<130000 THEN 1310
1240 IF V>129999 THEN 1340
1250 DISPLAY AT(15,5):"KEINE
 BERECHNUNG" :: GOTO 1890
1260 F=.22*V-926
1270 GOTO 1350
1280 H=(V-18000)/10000
```

```
1290 F=(((3.05*H-73.76)*H+69
5)*H+2200)*H+3034
1300 GOTO 1350
1310 M=(V-60000)/10000
1320 F=(((.09*M-5.45)*M+88.1
3)*M+5040)*M+20018
1330 GOTO 1350
1340 F=.56*V-14837
1350 IF SP=2 THEN F=F*2
1360 DISPLAY AT(9,1):"ZU VER
ST. EINK.:" :: DISPLAY AT(9,
20):INT(JG)
1370 DISPLAY AT(11,1):"EINKO
MMENSTEUER:" :: DISPLAY AT(1
1,20):INT(F)
1380 IF KZ=1 THEN Z=600 :: G
OTO 1440
1390 IF KZ=2 THEN Z=1560 ::
GOTO 1440
1400 IF KZ=3 THEN Z=3360 ::
GOTO 1440
1410 IF KZ=4 THEN Z=5160 ::
GOTO 1440
1420 IF KZ=5 THEN Z=6960 ::
GOTO 1440
1430 IF KZ=6 THEN Z=8760
1440 Y=F-Z
1450 X=.09*Y :: IF X<0 THEN
X=0
1460 DISPLAY AT(13,1):"KIRCH
ENSTEUER  :" :: DISPLAY AT(1
3,20):INT(X)
1470 AKV=((BJ/100)*KV)/13
1480 ARV=((BJ/100)*RV)/13
1490 AALV=((BJ/100)*ALV)/13
1500 SVGES=AKV+ARV+AALV
1510 DISPLAY AT(20,1):"SIND
SIE SOZIALVERSICHERUNGSPFLIC
HTIG........................
..................(J/N)"
1520 ACCEPT AT(22,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1530 IF A$="J" THEN 1550
1540 IF A$="N" THEN 1780
1550 DISPLAY AT(20,1):""
1560 DISPLAY AT(21,1):""
1570 DISPLAY AT(15,1):"KRANK
ENVERS.BEITRAG:"
1580 DISPLAY AT(15,20):INT(A
KV)
1590 DISPLAY AT(17,1):"RENTE
NVERS.BEITRAG :"
1600 DISPLAY AT(17,20):INT(A
RV)
1610 DISPLAY AT(19,1):"ARBEI
TSLOSENVERS.   :"
1620 DISPLAY AT(19,20):INT(A
ALV)
1630 NG=BJ-F-X-(SVGES*13)
1640 MG=NG/13
1650 DISPLAY AT(21,1):"MONAT
LICH NETTO:"
1660 DISPLAY AT(21,20):INT(M
G)
1670 DISPLAY AT(22,1):"-----
------------------------"
1680 DISPLAY AT(24,1):"WERTE
 AUSDRUCKEN....(J/N)"
1690 DISPLAY AT(15,25):"}" :
: DISPLAY AT(16,25):"}" :: D
ISPLAY AT(17,25):"}MTL"
1700 DISPLAY AT(18,25):"}" :
: DISPLAY AT(19,25):"}"
1710 ACCEPT AT(24,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1720 IF A$="J" THEN 1930
1730 IF A$="N" THEN 1740
1740 DISPLAY AT(24,1):"NEUE
BERECHNUNG.....(J/N)"
1750 ACCEPT AT(24,28)BEEP VA
LIDATE("JN")SIZE(1):A$
1760 IF A$="J" THEN CALL CLE
AR :: GOTO 570
1770 IF A$="N" THEN CALL CLE
AR :: STOP
1780 DISPLAY AT(20,1):""
1790 DISPLAY AT(21,1):""
1800 NG=BJ-F-X
1810 MG=NG/13
1820 DISPLAY AT(21,1):"MONAT
LICH NETTO:"
1830 DISPLAY AT(21,20):INT(M
G)
1840 DISPLAY AT(22,1):"-----
------------------------"
1850 DISPLAY AT(24,1):"WERTE
 AUSDRUCKEN  (J/N)"
1860 ACCEPT AT(24,26)BEEP VA
LIDATE("JN")SIZE(1):A$
1870 IF A$="J" THEN 1930
1880 IF A$="N" THEN 1890
1890 DISPLAY AT(24,1):"NEUE
BERECHNUNG...(J/N)"
1900 ACCEPT AT(24,26)BEEP VA
LIDATE("JN")SIZE(1):A$
1910 IF A$="J" THEN CALL CLE
AR :: GOTO 890
1920 IF A$="N" THEN CALL CLE
AR :: STOP
1930 OPEN #1:"PIO"
1940 FOR Z=1 TO 23
1950 FOR P=1 TO 30
1960 CALL GCHAR(Z,P,X)
1970 DRU$=SEG$(DRU$,1,P)&CHR
$(X)
1980 NEXT P
1990 PRINT #1:DRU$
2000 NEXT Z
2010 CLOSE #1
2020 GOTO 1740
```

```
10 REM********************
11 REM*                  *
12 REM*  KOMPOSITIONEN   *
13 REM*  Copyright  by   *
14 REM*                  *
15 REM* A.Saberschinsky  *
16 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*  Kass.-Rekorder  *
20 REM*                  *
23 REM* Speicherbelegung *
24 REM*    9172 Bytes    *
25 REM********************
26 REM
100 DIM F1(125),F2(125),F3(1
25),L(125)
110 CALL CLEAR
120 PRINT "*NOTENKOMPOSITION
SPROGRAMM*": : : : : : : : :
 : :"ENTWICKELT UND GESCHRIE
BEN  VON"
130 PRINT ".....ALEXANDER SA
BERSCHINSKY1985": : : :
140 CALL SOUND(100,110,0,100
0,0,10000,0)
150 FOR Z=1 TO 2500
160 NEXT Z
170 FLAG1=0
180 FLAG11=0
190 FLAG3=0
200 CALL CLEAR
210 PRINT ".........WAHLLIST
E..........=================
==========="
220 PRINT "................
...........BEDIENUNGSANLEITU
NG........1NOTENEINGABE.....
..........2"
230 PRINT "NOTENKORREKTUR...
..........3ABSPIELEN DER KOM
POSITION..4LADEN EINER KOMPO
SITION....5"
240 PRINT "SPEICHERN EINER K
OMPOSITON.6ENDE............
..........7": : : : : :
250 CALL KEY(0,K,S)
260 IF S=0 THEN 250
270 IF K<49 THEN 250
280 IF K>55 THEN 250
290 ON K-48 GOTO 300,690,115
0,1540,1610,1690,1770
300 CALL CLEAR
310 PRINT "MIT DIESEM PROGRA
MM KOENNEN SIE BELIEBIGE MEL
ODIEN EIN- GEBEN, DIE NACH E
INER EVEN-"
320 PRINT "TUELLEN KORREKTUR
 ANGEHOERT ODER GESPEICHERT
WERDEN.....KOENNEN. AUCH DAS
 LADEN "
330 PRINT "ALTER MELODIEN VO
N CASSETTE IST MOEGLICH."
340 PRINT "DEN TONWERT GEBEN
 SIE EIN,..INDEM SIE FUER DI
E ENT-.....SPRECHENDE NOTE D
IE TASTE.."
350 PRINT "AUF DER ALPHANUMM
ERISCHEN...TASTATUR DRUECKEN
. FUER.....DIE MITTLERE TONL
EITER IST"
360 PRINT "NUR DIE ENTSPRECH
ENDE TASTE ZU DRUECKEN, FUER
 DIE TIEFE TONLEITER ZUSAETZ
LICH DIE "
370 PRINT "TASTE 'FCTN' UND
FUER DIE...HOHE TONLEITER DI
E TASTE....'CTRL'. (Z.B.: HO
HES G -->..CTRL G.)": :
380 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 CALL CLEAR
420 PRINT "BITTE MERKEN SIE
SICH.......WEITERHIN FOLGEND
E ZU-......SAETZLICHE TASTEN
BELEGUNG...(EV. NOTIEREN):"
430 PRINT "..NOTE......!...T
ASTE.......=================
===========..C#, D&....!....
..5......."
440 PRINT "..D#, E&....!....
..6..........F#, G&....!....
..7..........G#, A&....!....
..8......."
450 PRINT "..A#, B&....!....
..9.......-----------------
-----------": :
460 PRINT "# = ERHOEHUNG EIN
ER NOTE....& = VERMINDERUNG
EINER NOTE": :
470 PRINT "FUER DIE TONLEITE
RWAHL BITTEWIEDER 'FCTN' BZW
. 'CTRL'...VERWENDEN.": : :
480 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
490 CALL KEY(0,K,S)
500 IF S=0 THEN 490
510 CALL CLEAR
520 PRINT "DIE TASTEN FUER D
IE NOTEN-..LAENGE SIND WIE F
OLGT BE-...LEGT:"
530 PRINT "NOTENLAENGE !...T
ASTE.......=================
===========..32STEL....!....
..1......."
540 PRINT "..16TEL.....!....
..2...........8TEL.....!....
..3...........4TEL.....!....
..4......."
```

```
550 PRINT "...4TEL. . ..!....
..5...........HALBE....!....
..6...........GANZE....!....
..7......."
560 PRINT "-----------------
-----------JEWEILS MIT 'SHIF
T' !!!!!!!!----------------
-----------": : :
570 PRINT ">BITTE EINE TASTE
 DRUECKEN<"
580 CALL KEY(0,K,S)
590 IF S=0 THEN 580
600 CALL CLEAR
610 PRINT "FUER EINE PAUSE D
RUECKEN DIEBITTE BEI DER NOT
ENHOEHEN-..WAHL EIN 'P' UND
DANN DIE"
620 PRINT "GEWUENSCHTE LAENG
E. ........BEI DER NOTENEING
ABE KOENNENSIE ZWISCHEN EIN-
, ZWEI-,"
630 PRINT "UND DREIKLANG (OD
ER AUCH BE-ENDIGUNG DER EING
ABEN) WAEH-LEN."
640 PRINT "WENN SIE WIEDERHO
LT (2).....AUS DER WAHLLISTE
 WAEHLEN,..WIRD DIE GESPEICH
ERTE KOMPO-"
650 PRINT "SITION GELOESCHT!
!!": : : : :">BITTE EINE TAS
TE DRUECKEN<"
660 CALL KEY(0,K,S)
670 IF S=0 THEN 660
680 GOTO 170
690 CALL CLEAR
700 A=0
710 A=A+1
720 PRINT "NOTENEINGABE:....
...........^^^^^^^^^^^^^":"-
>(1) EINKLANG..............-
>(2) ZWEIKLANG.............-
>(3) DREIKLANG.............-
>(4) ENDE DER EINGABE": :
730 CALL KEY(0,K,S)
740 IF S=0 THEN 730
750 IF K<49 THEN 730
760 IF K>52 THEN 730
770 ON K-48 GOTO 780,850,960
,1110
780 FLAG4=1
790 GOSUB 1920
800 CALL SOUND(L(A),F1(A),0)
810 F2(A)=F1(A)
820 F3(A)=F1(A)
830 IF FLAG3=1 THEN 170
840 GOTO 710
850 FLAG2=1
860 FLAG4=2
870 PRINT "-------- 1. KLANG
-----------"
880 GOSUB 1920
890 FLAG5=1
900 PRINT "-------- 2. KLANG
-----------"
910 GOSUB 1920
920 CALL SOUND(L(A),F1(A),0,
F2(A),0)
930 F3(A)=F2(A)
940 IF FLAG3=1 THEN 170
950 GOTO 710
960 FLAG2=1
970 FLAG4=3
980 FLAG6=1
990 PRINT "-------- 1. KLANG
-----------"
1000 GOSUB 1920
1010 FLAG2=1
1020 FLAG6=2
1030 PRINT "-------- 2. KLAN
G ----------"
1040 GOSUB 1920
1050 FLAG6=3
1060 PRINT "-------- 3. KLAN
G ----------"
1070 GOSUB 1920
1080 CALL SOUND(L(A),F1(A),0
,F2(A),0,F3(A),0)
1090 IF FLAG3=1 THEN 170
1100 GOTO 710
1110 IF FLAG12=1 THEN 170
1120 A=A-1
1130 M=A
1140 GOTO 170
1150 CALL CLEAR
1160 PRINT "KORREKTUR:......
............^^^^^^^^^^": :
1170 PRINT "WELCHE KORREKTUR
?..........->(1) LOESCHEN E
INER NOTE...->(2) ERGAENZEND
E NOTE......->(3) VERAENDERN
 EINER NOTE": : :
1180 CALL KEY(0,K,S)
1190 IF S=0 THEN 1180
1200 IF K<49 THEN 1180
1210 IF K>51 THEN 1180
1220 ON K-48 GOTO 1350,1230,
1460
1230 M=M+1
1240 INPUT "BITTE GEBEN SIE
DIE NOTE AN,DIE NACH DER EIN
ZUFUEGENDEN NOTE STEHEN WIRD
. ":N
1250 FOR Y=M TO N+1 STEP -1
1260 Q=Y-1
1270 F1(Y)=F1(Q)
1280 F2(Y)=F2(Q)
1290 F3(Y)=F3(Q)
1300 L(Y)=L(Q)
1310 NEXT Y
1320 A=N
1330 FLAG3=1
```

```
1340 GOTO 1510
1350 INPUT "BITTE GEBN SIE D
IE NUMMER...DER NOTE AN, DIE
 GELOESCHT..WERDEN SOLL. ":N
1360 FOR A=N TO M
1370 I=A+1
1380 F1(A)=F1(I)
1390 F2(A)=F2(I)
1400 F3(A)=F3(I)
1410 L(A)=L(I)
1420 NEXT A
1430 M=M-1
1440 GOTO 170
1450 GOTO 1510
1460 INPUT "BITTE GEBEN SIE
NUMMER DER..NOTE EIN, DIE SI
E KORRI-....GIEREN WOLLEN. "
:A
1470 PRINT
1480 IF A<1 THEN 1460
1490 IF A>M THEN 1460
1500 IF INT(A)<>A THEN 1460
1510 PRINT :".......|.......
....................v": :
1520 FLAG3=1
1530 GOTO 720
1540 CALL CLEAR
1550 PRINT "ABSPIELEN:......
............^^^^^^^^^^": : :
 : :
1560 FOR A=1 TO M
1570 CALL SOUND(L(A),F1(A),0
,F2(A),0,F3(A),0)
1580 PRINT "NOTENNUMMER: ";A
1590 NEXT A
1600 GOTO 170
1610 CALL CLEAR
1620 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,INPUT ,FIXED
1630 INPUT #1:M
1640 FOR A=1 TO M
1650 INPUT #1:L(A),F1(A),F2(
A),F3(A)
1660 NEXT A
1670 CLOSE #1
1680 GOTO 170
1690 CALL CLEAR
1700 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,OUTPUT,FIXED
1710 PRINT #1:M
1720 FOR A=1 TO M
1730 PRINT #1:L(A),F1(A),F2(
A),F3(A)
1740 NEXT A
1750 CLOSE #1
1760 GOTO 170
1770 CALL CLEAR
1780 PRINT "BITTE BEACHTEN S
IE, DASS BEIABSCHLUSS DES PR
OGRAMMS ALLEDATEN GELOESCHT
WERDEN!": : :
1790 PRINT "BITTE WAEHLEN SI
E:..........-->(1) ZURUECK Z
UR WAHLLISTE-->(2) ENDE DES
PROGRAMMS": : : : :
1800 CALL KEY(0,K,S)
1810 IF S=0 THEN 1800
1820 IF K<49 THEN 1800
1830 IF K>50 THEN 1800
1840 IF K=49 THEN 170
1850 CALL CLEAR
1860 PRINT "IN DIESEM FALL D
ARF ICH MICHBEI IHNEN FUER I
HR INTERESSEBEDANKEN UND MIC
H VERAB-....SCHIEDEN."
1870 PRINT "  ALEXANDER SABE
RSCHINKY '85": : : : : :
1880 CALL KEY(0,K,S)
1890 IF S=0 THEN 1880
1900 CALL CLEAR
1910 END
1920 PRINT "NOTENNR.:";A;"":
 :"KLANG(FREQUENZ)? "
1930 CALL KEY(0,K,S)
1940 IF S=0 THEN 1930
1950 IF K=131 THEN 2340
1960 IF K=132 THEN 2370
1970 IF K=133 THEN 2400
1980 IF K=134 THEN 2430
1990 IF K=135 THEN 2460
2000 IF K=129 THEN 2490
2010 IF K=136 THEN 2520
2020 IF K=67 THEN 2550
2030 IF K=68 THEN 2580
2040 IF K=69 THEN 2610
2050 IF K=70 THEN 2640
2060 IF K=71 THEN 2670
2070 IF K=65 THEN 2700
2080 IF K=72 THEN 2730
2090 IF K=96 THEN 2760
2100 IF K=9 THEN 2790
2110 IF K=11 THEN 2820
2120 IF K=123 THEN 2850
2130 IF K=125 THEN 2880
2140 IF K=124 THEN 2910
2150 IF K=191 THEN 2940
2160 IF K=159 THEN 2970
2170 IF K=181 THEN 3000
2180 IF K=182 THEN 3030
2190 IF K=183 THEN 3060
2200 IF K=158 THEN 3090
2210 IF K=57 THEN 3120
2220 IF K=53 THEN 3150
2230 IF K=54 THEN 3180
2240 IF K=55 THEN 3210
2250 IF K=56 THEN 3240
2260 IF K=15 THEN 3270
2270 IF K=14 THEN 3300
2280 IF K=12 THEN 3330
2290 IF K=1 THEN 3360
2300 IF K=6 THEN 3390
```

```
2310 IF K=80 THEN 3420
2320 IF K=83 THEN 32767
2330 GOTO 1920
2340 FRE=523
2350 C$="HOHES C"
2360 GOTO 3440
2370 FRE=587
2380 C$="HOHES D"
2390 GOTO 3440
2400 FRE=659
2410 C$="HOHES E"
2420 GOTO 3440
2430 FRE=698
2440 C$="HOHES F"
2450 GOTO 3440
2460 FRE=784
2470 C$="HOHES G"
2480 GOTO 3440
2490 FRE=880
2500 C$="HOHES A"
2510 GOTO 3440
2520 FRE=988
2530 C$="HOHES H"
2540 GOTO 3440
2550 FRE=262
2560 C$="MITTLERES C"
2570 GOTO 3440
2580 FRE=294
2590 C$="MITTLERES D"
2600 GOTO 3440
2610 FRE=330
2620 C$="MITTLERES E"
2630 GOTO 3440
2640 FRE=349
2650 C$="MITTLERES F"
2660 GOTO 3440
2670 FRE=392
2680 C$="MITTLERES G"
2690 GOTO 3440
2700 FRE=440
2710 C$="MITTLERES A"
2720 GOTO 3440
2730 FRE=494
2740 C$="MITTLERES H"
2750 GOTO 3440
2760 FRE=131
2770 C$="TIEFES C"
2780 GOTO 3440
2790 FRE=147
2800 C$="TIEFES D"
2810 GOTO 3440
2820 FRE=165
2830 C$="TIEFES E"
2840 GOTO 3440
2850 FRE=175
2860 C$="TIEFES F"
2870 GOTO 3440
2880 FRE=196
2890 C$="TIEFES G"
2900 GOTO 3440
2910 FRE=220
2920 C$="TIEFES A"
2930 GOTO 3440
2940 FRE=247
2950 C$="TIEFES H"
2960 GOTO 3440
2970 FRE=932
2980 C$="HOHES A#,B&"
2990 GOTO 3440
3000 FRE=554
3010 C$="HOHES C#,D&"
3020 GOTO 3440
3030 FRE=622
3040 C$="HOHES D#,E&"
3050 GOTO 3440
3060 FRE=740
3070 C$="HOHES F#,G&"
3080 GOTO 3440
3090 FRE=831
3100 C$="HOHES G#,A&"
3110 GOTO 3440
3120 FRE=466
3130 C$="MITTLERES A#,B&"
3140 GOTO 3440
3150 FRE=277
3160 C$="MITTLERES C#,D&"
3170 GOTO 3440
3180 FRE=311
3190 C$="MITTLERES D#,E&"
3200 GOTO 3440
3210 FRE=370
3220 C$="MITTLERES F#,G&"
3230 GOTO 3440
3240 FRE=415
3250 C$="MITTLERES G#,A&"
3260 GOTO 3440
3270 FRE=233
3280 C$="TIEFES A#,B&"
3290 GOTO 3440
3300 FRE=139
3310 C$="TIEFES C#,D&"
3320 GOTO 3440
3330 FRE=156
3340 C$="TIEFES D#,E&"
3350 GOTO 3440
3360 FRE=185
3370 C$="TIEFES F#,G&"
3380 GOTO 3440
3390 FRE=208
3400 C$="TIEFES G#,A&"
3410 GOTO 3440
3420 FRE=40000
3430 C$="PAUSE"
3440 PRINT TAB(7);C$;"": :
3450 ON FLAG4 GOTO 3460,3480
,3540
3460 F1(A)=FRE
3470 GOTO 3600
3480 IF FLAG5=1 THEN 3510
3490 F1(A)=FRE
3500 GOTO 3600
```

```
3510 F2(A)=FRE
3520 FLAG5=0
3530 GOTO 3600
3540 ON FLAG6 GOTO 3550,3570
,3590
3550 F1(A)=FRE
3560 GOTO 3600
3570 F2(A)=FRE
3580 GOTO 3600
3590 F3(A)=FRE
3600 IF FLAG2=1 THEN 3930
3610 PRINT "KLANGLAENGE? "
3620 CALL KEY(0,K,S)
3630 IF S=0 THEN 3620
3640 IF K=33 THEN 3720
3650 IF K=64 THEN 3750
3660 IF K=35 THEN 3780
3670 IF K=36 THEN 3810
3680 IF K=37 THEN 3840
3690 IF K=94 THEN 3870
3700 IF K=38 THEN 3900
3710 GOTO 3620
3720 L(A)=63
3730 C$="32-TEL"
3740 GOTO 3920
3750 L(A)=125
3760 C$="16-TEL"
3770 GOTO 3920
3780 L(A)=250
3790 C$=" 8-TEL"
3800 GOTO 3920
3810 L(A)=500
3820 C$=" 4-TEL"
3830 GOTO 3920
3840 L(A)=750
3850 C$=" 4-TEL PUNKTIERT"
3860 GOTO 3920
3870 L(A)=1000
3880 C$=" HALBE"
3890 GOTO 3920
3900 L(A)=2000
3910 C$=" GANZE"
3920 PRINT TAB(7);C$;: : :
3930 FLAG2=0
3940 RETURN
```

UNSER TELEFONSERVICE

Leserbriefe und Fragen sind uns stets willkommen. Wir beantworten sie entweder direkt oder auf der Leserbriefseite

ACHTUNG !!!

Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13

BÖRSE

Drucke ihre Listings für 6,– DM pro Prg. Entferne Listschutz für 3,– DM pro Cassette !! Info 50 Pfg. anfordern. Matthias Orf, Birkenallee 34, D-3507 Baunatal 1

TI99/4A + Ext. Basic + orig. Cas.Rec. Kabel + TI-Bücher + umfangr. Software für 500,– DM zu verk. Tel. 040/6103696

Werden Sie Mitglied im großen TI-CLUB BAUNATAL! Tolles Clubheft, Prg. Speicher, Programmierlenkurs u.v.m. Info 50 Pfg. oder Clubheft 3,– DM anfordern! TI-CLUB BAUNATAL (TCB), Matthias Orf, Birkenaalee 34, D-3507 Baunatal, Tel. 0561/497990 !!!

Verkaufe originalverpackt: TI99/4A (19 Monate): 235,– DM TI99/4A + Rec-Kabel: 250,– DM Modul: Parsec: 45,– DM, Modul: Munch-Man: 45,– TI-Revue 1-7 (1a Zust.): 25,– TI-Revue (1–7) einzeln: 4,–. Andreas Braun, Tel. 02473/7320

TI99/4A zu verkaufen!! TI99/4A (wenig gelaufen) 235,– DM (+ Rec-Kabel: 250,–). Spiele-Modul: Parsec 45,– DM – Anruf ab 17 Uhr bei Herman-Josef Wollgarten, Tel. 02473/8628

TI99/4A + XB + Recorder + Kabel + Datenv. + Schach + Spielmodule + Joyst. + Literatur + Software DM 550,– 08461/1404

TI99 / 4A + ExBas + Peribox + RS232 + 32K + Disk + Edi/Ass + TI-Writer + Div. Module + Literatur 2500 DM. Apfeldorf 04122/54697 ab 18 Uhr

---

TI99/4A Verkaufe super Programme in TI-Ex-Basic. Info an Frank Henneberger, Pf. Legemann 19a, Nörvenich 2

TI-Writer, DM 210, Pers. Record. Keeping DM 40 Pers. Report Generator DM 40,– Statistik DM 40,–. 64K Platine u. Anleitung n. TI-Reven DM 25,– Tel. 0511/35 63 132 n. 18 h

---

Wer hat Lust, mit Anfänger Programme zu tauschen (TI-B)? Tel. 05971/57124). Holger Hammerschlag, 4440 Rheine 1

TI99+2Tc Konsole + X-Bas+Rec. Kabel + Joy + div. Bücher 3 div. Progr. Cass. Preis VB 600 DM Schwetje 040/715 70 06

Suche Ex-Basic, Mini-Memory, ext. 32KB. Heinz Jörgensen, Ewensweg 2, 2987 Grossheide 1

TI-99/4A + Cass.-Rek. Kabel + Cass.-Rek. + Handbuch + Software Pries: VB 250,–. Tel: 02354/6974, 5882 Meinerzhagen 1

TI 99/4A: Verkaufe Buchungsjournal DM 70,– Bastert, Heinz Werner, 469 Herne 20, Tulpenweg 61

Suche Ext. Basic, evtl. mit Konsole, aber spottbillig und funktionstüchtig! Tel. 02103/44131

Suche komplette Peripherie, T. 09332/31 95

Tausche: HiFi-Cassettendeck Nakamichi-„Dragon" Reference bei AUDIO und STEREO PLAY Testberichte 8/83 Audio + 3/83 Stereoplay. Gegen: kompl. ausgestattete TI-99/4A Computer-Anlage. Mindest-Wert sollte nicht DM 3.500,– unterschreiten. Angebote an: R. Göhring, Alvesloher Str. 21, 2358 Kaltenkirchen, Tel. 04191/73 05

SPIEL SALON



See-Schlacht

Wir haben das Spiel „See-Schlacht" genannt, obwohl es sich eigentlich um ein U-Boot Spiel handelt! (es existieren bereits mehrere Spiele mit dem Namen „U-Boot") Das Spiel ist für zwei Personen gedacht, die ihre beiden Schiffe mit den Joysticks 1 und 2 steuern (geübten Programmierern wird es zwar nicht schwerfallen, das Spiel auf die Konsolentastatur umzuschreiben, was aber den Nachteil hat, daß zwei Spieler unabhängig voneinander die Tastatur bedienen müssen).

Die Aufgabe jedes Spielers besteht darin, den Gegner abzuschießen. Das Unterseeboot ist in jede Richtung steuerbar, genauso wie es in jede Richtung schießen kann. Geschossen wird, indem man auf den Feuerknopf drückt und innerhalb von ca. drei Sekunden den Joystick in die gewünschte Schußrichtung drückt. Besonders reizvoll ist es auch, wenn man Gänge in die Felswand schießt, um sich darin vor den Schüssen des Gegners zu schützen. Die Schüsse haben nur eine gewisse Reichweite und es kommt deshalb auf eine gute Reaktion im Nahkampf an. Das Anzeigenfeld rechts oben zeigt die restlichen Schüsse und die Siege des jeweiligen Spielers an. Gute Reaktion wird auch am Anfang verlangt, da es zu Spielbeginn nicht bekannt ist, wer welches Schiff zu steuern hat.

Zum Programm:

Um die einzelnen Steuerroutinen nicht doppelt schreiben zu müssen, haben wir eine zweidimensionale Variable verwendet, die angibt, welcher Spieler am Zuge ist. Dies bedeutet aber nicht, daß ein Joystick bevorzugt bzw. benachteiligt ist, da diese Variable laufend geändert wird.

Das Programm wurde möglichst einfach gehalten, um der Schnelligkeit des doch relativ langsamen TI Basic nichts zu nehmen.

Das Programm ist zusätzlich noch mit einigen REMarks erklärt. Das Spiel kann auch mit dem Xbasicmodul gefahren werden und wird dadurch wesentlich schneller.

Rühner + Rühmer

Apfelfänger

Haben Sie schon einmal in einer Fabrik als Apfelfänger gearbeitet? Nein? Dann versuchen Sie es doch mal mit Ihrem TI in der Grundversion und diesem Programm.

Sie müssen versuchen, die Äpfel, die übers Fließband laufen und irgendwo vom Band fallen, unten mit dem Korb aufzufangen. Ist Ihnen dieses gelungen, laufen Sie schnell zu einer der beiden Tonnen; rechts und links; und lassen Sie sich für den gefangenen Apfel Ihren Lohn gutschreiben. Als ordentlicher Fabrikarbeiter werden Sie natürlich nach dem jeweils gültigen Tarif bezahlt. Wenn Sie 10 Äpfel in der vorgegebenen Zeit zur Tonne gebracht haben, dürfen Sie am nächsten Tag weiterarbeiten. Ihr Tarif erhöht sich von Tag zu Tag um 50 %; die Zeit aber verkürzt sich jeden Tag um etwa 10 %.

Sollten Sie an einem Tag mit der Zeit nicht hinkommen, werden Sie mit Ihrem Lohn entlassen.

Steuern können Sie den Korb mit folgenden Tasten:

‚A' links
‚D' rechts
‚B' Funktionstaste

Die Taste ‚B' hat zwei Funktionen:

1. Wenn ein Apfel in der Luft ist, kann damit der Fall beschleunigt werden, wodurch man Zeit sparen kann. Der Nachteil ist aber, daß man den Korb nicht mehr bewegen kann, bis der Apfel unten ist. Daher sollte man schon sehr sicher sein, daß man unter dem Apfel steht, wenn man die Taste ‚B' benutzt.
2. Wenn kein Apfel in der Luft ist, kann man mit Taste ‚B' den nächsten Apfel kommen lassen.

Sollten Sie mehr als 4 Tage in der Fabrik aushalten, können Sie schon stolz auf Ihre Leistung sein.

Viel Spaß mit diesem Spiel wünscht

Torsten Rauer

Variable:

Die wichtigsten Variablen:
Q = die feste Zeit
F = Anzahl der noch zu fangenden Äpfel
G = Tarif
E = Guthaben
D = Noch vorhandene Zeit
B = 3 mögliche Positionen des fallenden Apfels

```
10 REM********************
11 REM*                  *
12 REM*  APFELFAENGER    *
13 REM*  Copyright  by   *
14 REM*  Thorsten Rauer  *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*    4724 Bytes    *
25 REM********************
26 REM
100 REM ***VORBEREITUNGEN
110 DIM B(3)
120 CALL CLEAR
130 CALL SCREEN(8)
140 FOR I=1 TO 12
150 CALL COLOR(I,2,8)
160 NEXT I
170 CALL COLOR(1,3,8)
180 CALL COLOR(9,5,8)
190 CALL COLOR(11,7,8)
200 CALL COLOR(12,14,8)
210 CALL CHAR(98,"00386CC6D6
C66C38")
220 CALL CHAR(106,"000000001
818FFFF")
230 CALL CHAR(115,"1A0C081C3
66E7E3C")
240 CALL CHAR(122,"387C44545
454447C")
250 CALL CHAR(107,"384444FEF
E7C7C38")
260 CALL CHAR(35,"FFFFFFFFFF
FFFFFF")
270 A=14
280 Q=700
290 D=Q
300 F=10
310 G=25
320 REM ***TITELBILD
330 PRINT : : : : : : :" s s
 s s s s s s s s s s s": : :
" s        APFELFAENGER     s"
: : :" s s s s s s s s s s s
 s s": : :
340 FOR I=1 TO 3
350 CALL SOUND(200,550,14,57
5,12)
360 CALL SOUND(200,550,12,57
5,10)
370 CALL SOUND(200,550,10,57
5,8)
380 CALL SOUND(800,210,4,221
,2)
390 NEXT I
400 PRINT ".s...c.THORSTEN.R
AUER....s": : :
410 PRINT " s s s s s s s s
s s s s s":"jjjjjjjjjjjjjjjj
jjjjjjjjjjjj":"b....b....b..
..b....b....b": : :
420 PRINT " BITTE EINE TASTE
 DRUECKEN"
430 CALL KEY(0,K,S)
440 CALL SCREEN(14)
450 CALL SCREEN(3)
460 CALL SOUND(50,876,17,863
,19)
470 IF S=0 THEN 430
480 REM  ***SPIELAUFBAU
490 CALL CLEAR
500 FOR I=1 TO 13
510 READ AA,AB,AC,AD
520 CALL HCHAR(AA,AB,AC,AD)
530 NEXT I
540 FOR I=1 TO 2
550 READ A$,L,N
560 GOSUB 640
570 NEXT I
580 GOSUB 690
590 GOTO 840
600 DATA 1,1,35,96,5,1,106,3
2,6,1,98,1,6,5,98,1,6,9,98,1
,6,13,98,1,6,17,98,1,6,21,98
,1
610 DATA 6,25,98,1,6,29,98,1
,21,3,122,1,21,30,122,1,22,1
,35,96
620 DATA ZEIT######S,2,4,GUT
HABEN######DM###TARIF####DM,
23,2
630 REM ***PRINTROUTINE
640 FOR SS=1 TO LEN(A$)
650 CALL HCHAR(L,N-1+SS,ASC(
SEG$(A$,SS,1)))
660 NEXT SS
670 RETURN
680 REM ***ZAEHLER
690 CALL HCHAR(2,9,35,5)
700 FOR SS=1 TO LEN(STR$(D))
710 CALL HCHAR(2,8+SS,ASC(SE
G$(STR$(D),SS,1)))
720 NEXT SS
730 CALL HCHAR(2,17,35,16)
740 CALL HCHAR(2,20,115,F)
750 CALL HCHAR(23,11,35,5)
760 FOR SS=1 TO LEN(STR$(E))
770 CALL HCHAR(23,10+SS,ASC(
SEG$(STR$(E),SS,1)))
780 NEXT SS
790 CALL HCHAR(23,27,35,3)
800 FOR SS=1 TO LEN(STR$(G))
810 CALL HCHAR(23,26+SS,ASC(
SEG$(STR$(G),SS,1)))
820 NEXT SS
830 RETURN
840 REM ***BEGINN
850 CALL HCHAR(21,A,107)
```

```
860 O=0
870 RANDOMIZE
880 B(2)=INT(RND*18+5)
890 B(1)=B(2)-1
900 B(3)=B(2)+1
910 FOR I=1 TO B(2)
920 CALL HCHAR(5,I,115)
930 CALL SOUND(3,110,0,112,0
,1110,3)
940 CALL HCHAR(5,I,106)
950 NEXT I
960 FOR I=0 TO 30 STEP 5
970 CALL SOUND(-300,750,I)
980 NEXT I
990 REM***BEWEGUNGSABLAUEFE
1000 FOR I=7 TO 20
1010 IF O=1 THEN 1030
1020 R=INT(RND*3+1)
1030 CALL HCHAR(I,B(R),115)
1040 CALL SOUND(20,140,3,-1,
0)
1050 IF O<>0 THEN 1160
1060 CALL KEY(1,K,S)
1070 D=D-1
1080 IF (K=1)+(K=3)+(K=16)TH
EN 1090 ELSE 1160
1090 IF K<>16 THEN 1120
1100 O=1
1110 GOTO 1160
1120 CALL HCHAR(21,A,32)
1130 A=A+(K-2)
1140 IF (A<4)+(A>29)THEN 140
0
1150 CALL HCHAR(21,A,107)
1160 CALL HCHAR(I,B(R),32)
1170 NEXT I
1180 GOSUB 680
1190 IF A=B(R)THEN 1350
1200 FOR I=550 TO 110 STEP -
20
1210 CALL SOUND(-300,I,2)
1220 NEXT I
1230 CALL KEY(1,K,S)
1240 D=D-1
1250 IF D<=0 THEN 1760
1260 IF D>40 THEN 1280
1270 CALL SOUND(35,342,0)
1280 IF (K=1)+(K=3)+(K=16)TH
EN 1290 ELSE 1230
1290 IF K=16 THEN 850
1300 CALL HCHAR(21,A,32)
1310 A=A+(K-2)
1320 IF (A<4)+(A>29)THEN 140
0
1330 CALL HCHAR(21,A,107)
1340 GOTO 1230
1350 FOR AA=0 TO 30 STEP 10
1360 CALL SOUND(-400,1450,AA
)
1370 NEXT AA
```

```
1380 P=1
1390 GOTO 1230
1400 A=A-(K-2)
1410 IF P=1 THEN 1440
1420 IF I<20 THEN 1150 ELSE
1330
1430 REM ***TONNE MIT APFEL
        ***ERREICHT
1440 FOR I=585 TO 1150 STEP
25
1450 CALL SOUND(25,I,0)
1460 NEXT I
1470 E=E+G
1480 F=F-1
1490 P=0
1500 GOSUB 690
1510 IF F=0 THEN 1520 ELSE 1
330
1520 CALL SOUND(600,667,0,68
7,0)
1530 A$="PRIMA!DU HAST WIEDE
R"
1540 L=11
1550 N=6
1560 GOSUB 640
1570 A$="EINEN TAG GESCHAFFT
 !"
1580 L=13
1590 N=6
1600 P=0
1610 O=0
1620 GOSUB 640
1630 G=G+(G/100*50)
1640 IF G<>INT(G)THEN 1650 E
LSE 1670
1650 G=INT(G+1)
1660 E=E+(INT(D))
1670 Q=Q-(Q/100*10)
1680 D=Q
1690 F=10
1700 GOSUB 690
1710 CALL KEY(0,K,S)
1720 IF S=0 THEN 1710
1730 CALL HCHAR(11,1,32,32)
1740 CALL HCHAR(13,1,32,32)
1750 GOTO 850
1760 REM ***ENDE
1770 FOR I=1 TO 30 STEP 3
1780 CALL SOUND(150,110,I,35
0,I)
1790 NEXT I
1800 CALL CLEAR
1810 PRINT : : : : : : : : :
 :" LEIDER HABEN SIE AN DIES
EM": :" TAG NICHT SCHNELL GE
NUG": :" GEARBEITET.": :
1820 PRINT " DAHER MUESSEN W
IR SIE ": :" MIT IHREM BISHE
R ": :" ERARBEITETEN KAPITAL
 VON": :" ";E;"DM ENTLASSEN.
```

```
": :
1830 CALL KEY(0,K,S)
1840 CALL SCREEN(16)
1850 CALL SCREEN(9)
1860 CALL SOUND(25,110,0,110
0,0)
1870 IF S=0 THEN 1830
1880 CALL CLEAR
1890 PRINT " MOECHTEN SIE SI
CH ": :" NOCHEINMAL BEI UNS
BEWERBEN": : :" WENN JA DANN
 'J' DRUECKEN": : : :
1900 CALL KEY(0,K,S)
1910 CALL SCREEN(4)
1920 CALL SCREEN(5)
1930 CALL SOUND(25,110,5,-1,
0)
1940 IF S=0 THEN 1900
1950 IF K=74 THEN 1980
1960 CALL CLEAR
1970 END
1980 P=0
1990 O=0
2000 RESTORE 600
2010 GOTO 100
```

BÖRSE

GPL-Disassembler für XB+32K+Disk mit ausf. Anl. 20 DM. Deutsche Erklärung aller Forth-Wörter 20 DM. Suche Anleitung zum SST-Basic-Compiler. R. Prinzensing, Geitlingstr. 27, 4630 Bochum 6

Maschinenprogramme für XB, MM u. E/A: GPL-Disassembler, Fast-Copy etc. Info gg. Rückumschlag: Eichhorn, Mathias, Ziegelheck 1, 6240 Königstein

Tausche oder verkaufe Prog. in X-Basic oder Assembler. Überwiegend Anwenderprog. / Utilitys. Manfred Rüß, Tel. 030/363 26 55

LOGO-Interpreter ermögicht tolle Grafik Spielereien (XB) 20 DM an M. Bannert – Mozartstr. 5 8676 Schwrzenbach/Saale. Tel: 09284–8338

Verk. TI-Kochbuch 1 = 2 = DM 40,– TI-Tips & Tricks = DM 40. Joystickadapter = DM 25. M. Bannert. Tel. 09284-8338

Verk. TI99/4A + Ex-Basic + Parsec + Pec. Kabel + Flougsimulator + Prog. + Listings nur 495 DM. Neupr. 800 DM. Tel. 02671/14583

Ti99/4A + Ex-Basic + Rec-Kabel + Haushalts- und Schachmodul + 2 Bücher, VB 650 DM, nur kompl. Tel. 0211/55 45 09, nach 16 Uhr

TI-Module, Bücher, Drucker, Programme. 3S-Liste geg. Freiumschalg. Ralph Schad, Wasserbett 6, 68 Mannheim 51

Suche für den Ti: Spielmodule, Laufwerk, 32k Erweiterung, Schnittstelle extern oder wenn günstig in (mit) Peri-box. 0571/71597

```
10 REM********************
11 REM*                  *
12 REM*   SEE-SCHLACHT   *
13 REM*   Copyright by   *
14 REM* Ruhmer & Ruhmer  *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*   Joystick 1&2   *
20 REM*                  *
21 REM*                  *
23 REM* Speicherbelegung *
24 REM*    5548 Bytes    *
25 REM********************
26 REM
220 CALL COLOR(12,16,10)
230 CALL CHAR(122,"F84444487
6514AC4")
240 CALL CLEAR
250 PRINT " ~~~.............
............~z~ - Ruhmer Sof
t...........~~~": :".......p
resents": : : : :
260 PRINT "*****************
***********......See-Schlac
ht........*****************
***********": : : :
270 PRINT "COPYRIGHT BY Ruhm
er Manfred ": :"...........&
 Ruhmer Klaus"
280 GOSUB 1870
290 CALL KEY(1,K,T1)
300 CALL KEY(2,K,T)
310 IF (T=-1)+(T1=-1)THEN 31
5 ELSE 290
315 GOSUB 1950
320 CALL CLEAR
330 CALL CHAR(97,"FFFFFFFFFF
FFFFFF")
340 CALL CHAR(98,"80C0E0E0F0
FCFEFF")
350 CALL CHAR(99,"80C0C0C0C0
E0FEFF")
360 CALL CHAR(100,"FF7F3F3F1
F0F0701")
370 CALL CHAR(101,"FFFEFEFCF
CFCF080")
380 CALL CHAR(102,"01073F3F7
F7F7FFF")
390 CALL CHAR(120,"")
400 CALL CHAR(111,"3C3C3C7E9
9997E3C")
410 CALL CHAR(110,"000000181
8")
420 CALL CHAR(112,"0000001F1
0101010")
430 CALL CHAR(113,"000000F01
0101010")
440 CALL CHAR(114,"101010F0"
)
```

```
450 CALL CHAR(115,"1010101F"
)
460 CALL CHAR(116,"10101010l
0101010")
470 CALL CHAR(117,"000000FF"
)
480 CALL CHAR(118,"1010101F1
0101010")
490 CALL CHAR(119,"101010F01
0101010")
500 SG(1)=0
510 SG(2)=0
520 CALL COLOR(10,16,6)
530 CALL COLOR(12,1,1)
540 CALL COLOR(8,3,3)
550 CALL COLOR(9,8,6)
560 CALL COLOR(1,6,6)
570 FOR N=1 TO 2
580 SU(N)=15
590 SH(N)=3
600 NEXT N
610 Z(1)=2
620 Z(2)=5
630 Z(3)=3
640 Z(4)=6
650 PRINT "^^^^^^^^^^^^^^^^^
^^^^puuuuuuaaaaaaaaaaaaaaaa
aaa^tPL.1:ie...daae.........
.da^tSIEGE:"
660 PRINT ".....daac....fc..
..a^vuuuuuub.....daaaaaaaab.
..a^tPL.2:lab.....daaaaaaaac
..a^tSIEGE:"
670 PRINT "aac.....de....daa
b a^suuuuuuaaa...........da
e a^^^^^^^^aaab............
..daaaaaaaa"
680 PRINT "aaae.............
.....daaae aae..............
......daac ae.....fab....fab
.......aaab"
690 PRINT "......faaaab faaa
.......daaa.....faaaaaaaaaae
........daa.....dae..daaaae.
...fc......"
700 PRINT "................
..faab......................
..aaaaaaaaa.................
..daaaaaaaa"
710 PRINT "........fb.......
...daaaaaaa......faab..fac.
......daae b.....faaaacfaae.
........da.."
720 PRINT "ac...faaaaaaaae..
..........faaaaaaaaaaaaaabfa
aaaaaaaaaaa"
730 CALL HCHAR(24,2,94,30)
740 CALL VCHAR(1,1,94,24)
750 CALL VCHAR(1,2,94)
760 CALL VCHAR(2,2,97,22)
770 CALL HCHAR(1,31,117)
780 CALL HCHAR(4,31,117)
790 CALL HCHAR(7,31,117)
800 CALL HCHAR(8,31,94)
810 CALL VCHAR(1,32,116,7)
820 CALL VCHAR(8,32,94,17)
830 CALL VCHAR(9,31,97,15)
840 CALL HCHAR(2,31,53)
850 CALL HCHAR(5,31,53)
860 CALL HCHAR(1,32,113)
870 CALL HCHAR(4,32,119)
880 CALL HCHAR(7,32,114)
890 RANDOMIZE
900 M=0
910 N=INT(RND*2)+1
920 M=M+1
930 A(N)=INT(RND*28)+3
940 B(N)=INT(RND*20)+3
950 CALL GCHAR(B(N),A(N),C)
960 IF C<>32 THEN 930
970 CALL HCHAR(B(N),A(N),111
)
980 A$=STR$(SG(N))
990 SP=31
1000 ZE=Z(N+2)
1010 GOSUB 1660
1020 GOSUB 1710
1030 IF M=1 THEN 920
1040 REM  * HAUPTPROGRAMM *
1050 CALL JOYST(N,X(N),Y(N))
1060 CALL KEY(N,K,T)
1070 IF K<>18 THEN 1100
1080 GOSUB 1200
1090 GOTO 1170
1100 IF (ABS(X(N))=0)*(ABS(Y
(N))=0)THEN 1170
1110 CALL GCHAR(B(N)-(Y(N)/4
),A(N)+(X(N)/4),C)
1120 IF C=32 THEN 1130 ELSE
1170
1130 CALL HCHAR(B(N)-(Y(N)/4
),A(N)+(X(N)/4),111)
1140 CALL HCHAR(B(N),A(N),32
)
1150 B(N)=B(N)-(Y(N)/4)
1160 A(N)=A(N)+(X(N)/4)
1170 GOSUB 1710
1180 GOTO 1050
1190 REM SCHUSSUNTERPROGRAMM
1200 R(N)=A(N)
1210 S(N)=B(N)
1220 FOR I=1 TO 16
1230 CALL JOYST(N,X(N),Y(N))
1240 IF (ABS(X(N))=4)+(ABS(Y
(N))=4)THEN 1270
1250 NEXT I
1260 RETURN
1270 ZE=Z(N)
1280 SP=30
1290 SU(N)=SU(N)-1
1300 IF SU(N)<0 THEN 1540
```

```
1310 CALL SOUND(100,-6,0)
1320 A$=STR$(SU(N))
1330 IF LEN(A$)=2 THEN 1350
1340 A$=STR$(SU(N))&CHR$(120
)
1350 GOSUB 1660
1360 FOR I=1 TO 5
1370 CALL GCHAR(S(N)-(Y(N)/4
),R(N)+(X(N)/4),C)
1380 IF C=94 THEN 1260
1390 CALL HCHAR(S(N)-(Y(N)/4
),R(N)+(X(N)/4),110)
1400 S(N)=S(N)-(Y(N)/4)
1410 R(N)=R(N)+(X(N)/4)
1420 CALL HCHAR(S(N),R(N),32
)
1430 IF C<>111 THEN 1490
1440 GOTO 1530
1450 CALL HCHAR(B(N),A(N),32
)
1460 GOSUB 1710
1470 CALL HCHAR(B(N),A(N),32
)
1480 GOTO 890
1490 IF C<>32 THEN 1260
1500 NEXT I
1510 GOTO 1260
1520 REM * AUSWERTUNG *
1530 GOSUB 1710
1540 SH(N)=SH(N)-1
1550 GOSUB 1870
1560 IF SH(N)>0 THEN 1630
1570 GOSUB 1710
1580 SG(N)=SG(N)+1
1590 GOSUB 1870
1600 GOSUB 1870
1610 IF SG(N)>2 THEN 1760
1620 GOTO 570
1630 GOSUB 1710
1640 GOTO 1450
1650 REM * SCHREIBROUTINE *
1660 FOR I=1 TO LEN(A$)
1670 CALL HCHAR(ZE,SP,ASC(SE
G$(A$,I,1)))
1680 SP=SP+1
1690 NEXT I
1700 RETURN
1710 IF N=1 THEN 1740
1720 N=1
1730 GOTO 1750
1740 N=2
1750 RETURN
1760 CALL CLEAR
1770 FOR I=1 TO 8
1780 CALL COLOR(I,16,8)
1790 NEXT I
1800 CALL SCREEN(8)
1810 PRINT "PL.";N;"HAT GEWO
NNEN !!!":"-----------------
-----": : : : :
1820 PRINT "NOCH EIN SPIEL G
EGEN DEN ": :"CHAMPION ? (J/
N)"
1830 CALL KEY(0,K,T)
1840 IF (K=74)+(K=106)THEN 5
00
1850 IF T=0 THEN 1830
1860 END
1870 CALL SOUND(120,262,8)
1880 CALL SOUND(70,110,30)
1890 CALL SOUND(120,262,8,33
0,6)
1900 CALL SOUND(70,110,30)
1910 CALL SOUND(120,262,7,33
0,5,392,3)
1920 CALL SOUND(70,110,30)
1930 CALL SOUND(450,330,3,39
2,2,523,0)
1940 RETURN
1950 CALL CLEAR
1960 PRINT "...*** SPIELANLE
ITUNG ***": : :"SIE BEFINDEN
 SICH TIEF UNTER": :"DER WAS
SEROBERFLAECHE IN": :
1970 PRINT "EINEM U-BOOT DAS
 SIE MIT": :"DEM JOYSTICK ST
EUERN MUESSEN": :"ES SPIELEN
 2 SPIELER DIE": :
1980 PRINT "IHRE U-BOOTE JEW
EILS MIT": :"JOYSTICH 1 ODER
 JOYSTICK 2": :"STEUERN"
1990 PRINT : :"T A S T E  D
R U E C K E N "
2000 CALL KEY(1,T,U)
2010 CALL KEY(2,T1,U1)
2020 IF (U=0)*(U1=0)THEN 200
0
2030 CALL CLEAR
2040 PRINT "SIE MUESSEN NUN
VERSUCHEN": :"DEN GEGNER ABZ
USCHIESSEN": :"GESCHOSSEN WI
RD INDEM MAN": :
2050 PRINT "AUF DIE ACTION-T
ASTE DRUECKT": :"UND SOFORT
DEN JOYSTICK IN": :"DIE GEWU
ENSCHTE SCHUSS-": :
2060 PRINT "RICHTUNG DRUECKT
": : :".....*** VIEL SPASS *
**": :"T A S T E  D R U E C
K E N"
2070 CALL KEY(1,T,U)
2080 CALL KEY(2,T1,U1)
2090 IF (U=0)*(U1=0)THEN 207
0
2100 CALL CLEAR
2110 RETURN
```

BUNTE MISCHUNG:

Pokelisten-generator

Das Programm dient zur Erstellung von Pokelisten auf einem Drucker. Es fragt zunächst die Schnittstellenbezeichnung sowie Anfangs- und Endadresse des gewünschten Bereiches ab. Anschließend wird die Pokeliste in DATA-Zeilen ausgedruckt. Ein im Speicher befindliches Assembler-Programm kann dadurch in eine abschreibfähige Form gebracht und später in einem BASIC-Programm verwendet werden.

Karnaugh-Veitch-Diagramm

Dieses Programm dient zur Ermittlung von Schaltfunktionen aus einer Wahrheitstabelle. Dabei wird zunächst eine Wahrheitstabelle für vier Variable erstellt und die Eingabe des jeweiligen Zustandes für A abgefragt. Bei einer fehlerhaften Eingabe kann durch Betätigen der Taste K der zuletzt eingegebene Wert korrigiert werden. Mit x werden die sogenannten don't care Felder gekennzeichnet. Sie werden dann verwendet, wenn der Zustand des Ausgangs A für die Berechnung der Schaltfunktion ohne Bedeutung ist. Nach Abschluß des Eingabevorganges erstellt das Programm das sogenannte KV-Diagramm, indem es die Werte für A bestimmten Feldern des Diagramms zuordnet. Das KV-Diagramm ist die grafische Veranschaulichung der Wahrheitstabelle einer Schaltfunktion. Dieses Verfahren der Vereinfachung von Schaltfunktionen wurde 1952 von E.W. Veitch erfunden und 1953 von M. Karnaugh weiterentwickelt. Das Programm ermittelt nun aus dem Diagramm die Schaltfunktion, indem es mit 1 gekennzeichnete Felder zu Gruppen zusammenfaßt und diesen ein bestimmtes Ergebnis zuordnet.

Dieses Programm kann in zwei Arten angewendet werden:
Zum einen können vorhandene Schaltungen minimiert werden. Dazu wird aus der vorhandenen Schaltung die Wahrheitstabelle ermittelt und eingegeben. Für Eingangskombinationen, die nicht vorkommen oder nicht von Bedeutung sind, wird x eingegeben. Das Programm ermittelt aus der Eingabe die optimierte Schaltfunktion, aus der die Schaltung entworfen werden kann.

Zum anderen kann aus einer entwickelten Wahrheitstabelle die minimierte Schaltfunktion ermittelt werden. Die Eingabe erfolgt analog zum o.a. Beispiel.

Bernhard Schmidt

Hexdump

Mit diesem Programm kann der Speicherinhalt auf einem Drucker ausgegeben werden. Es werden jeweils 16 Werte in hexadezimaler Form und die Hex-Adresse des jeweils ersten Wertes ausgegeben.

Bernard Schmidt

Widerstand

Das Programm „Widerstand" ist für den Elektronik-Bastler gedacht.

Im Menue kann folgende Auswahl getroffen werden.
0 – Errechnen von parallelgeschalteten Widerständen zu einem Ersatzwiderstand 2, 3, 4 und 5 parallele können berechnet werden.
1 und 2 berechnet der Farbcode
z.B. 0 = schwarz oder rot = 2
3. 4 u. 5 errechnen wahlweise Widerstand, Strom oder Spannung.
6. ist das Programmende.

Rüdiger Hussy

Space Invaders

Der alte Spielhallenhit „Space Invaders" hat nun auch endlich den TI99 eingeholt. Zwar eine etwas abgeänderte Version, auch etwas leichter zu spielen, aber unverkennbar standen die Original-Invaders für dieses Programm Pate. Zu den Spielregeln gibt es da wohl nicht mehr viel zu sagen. Jeder Computerfreak ist mit diesem Oldie vertraut. Daß wir es trotzdem abdrucken, hat drei Gründe. Erstens hat sich der Autor viel Mühe damit gemacht und das soll natürlich auch belohnt werden, denn dies ist Eure Zeitschrift und wir weisen nur manchmal ein bißchen in die Richtung, wo's lang gehen sollte. Zweitens ist das Spiel in TI-Basic geschrieben worden. Damit denken wir an die User, die („immer noch?") kein Extended-Basic Modul besitzen (Jetzt wird's aber langsam Zeit). Und drittens ist die Grafik ganz toll geworden. Auch die Spielgeschwindigkeit ist für die mageren Möglichkeiten des TI-Basic hervorragend programmiert. Zum Spielen nehmt Ihr übrigens Euren Joystick aus der Schublade. Damit läßt sich die Kanone nach rechts und links bewegen.

Prognose-Programm

Sinn: Dieses Programm errechnet aufgrund von max. 22 feststehenden Werten bis zu 22 Werte in die Zukunft (Prognose oder Hochrechnung).

Diese Werte werden auf dem Bildschirm ausgegeben. Anschließend werden diese Hochrechnungswerte auf einer Grafik dargestellt.
100–160 Dimensionierung und Titel
170 Einleitung
180–300 Abfrage der feststehenden Werte
310–430 Ausgabe der Prognosewerte (Hochrechnung)
440–820 Ausgabe der Grafik
830–900 Ende des Programms
910–950 Unterprogramm Stotterdruck

Bitte lesen Sie weiter auf Seite 66

```
1Ø ! *********************
11 ! *                   *
12 ! *     PROGNOSE      *
13 ! *                   *
14 ! *   Copyright by    *
15 ! *                   *
16 ! *     R. Bitter     *
17 ! *                   *
18 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic    *
22 ! *                   *
26 ! *  Speicherbelegung *
27 ! *     4447 Bytes    *
28 ! *                   *
29 ! *********************
3Ø !
1ØØ CALL CLEAR :: DIM W(22),
DAT(22),WE(22):: CALL CHAR(2
3,"ØØØØØØFFFF")
11Ø W$="PROGNOSE-PROGRAMM"
12Ø CALL STOTT(12,W$)
13Ø ZE1=11 :: SP1=8 :: ZE2=1
3 :: SP2=26
14Ø CALL RAHMEN(ZE1,SP1,ZE2,
SP2)
15Ø DISPLAY AT(24,3):"progra
mmiert von r.bitter"
16Ø FOR I=1 TO 4ØØ :: NEXT I
17Ø CALL EINLEITUNG
18Ø CALL CLEAR
19Ø DISPLAY AT(1,1):"WIEVIEL
 WERTE LIEGEN VOR:" :: DISPL
AY AT(2,1):"MAX. 22 WERTE" :
: ACCEPT AT(1,22):N :: CALL
HCHAR(2,3,33,28)
2ØØ IF N>22 THEN 19Ø
21Ø FOR I=1 TO N
22Ø DISPLAY AT(I+2,3):"WERT"
 :: DISPLAY AT(I+2,8):USING
"##":I :: DISPLAY AT(I+2,1Ø)
:":" :: ACCEPT AT(I+2,15):W(
I)
23Ø NEXT I
24Ø T=LOG(W(1))
25Ø X=Ø
26Ø FOR I=2 TO N :: L=LOG(W(
I)):: T=T+L :: X=X+(I-1)*L :
: NEXT I
27Ø A=6*(2*X/(N-1)-T)/N/(N+1
)
28Ø G=EXP(A)-1 :: G=1ØØ*(EXP
(A)-1):: G=INT(G*1ØØ+.5)/1ØØ
29Ø DISPLAY AT(24,1):"MITTL.
WACHSTUMSRATE:" :: DISPLAY A
T(24,22):USING "###.##":G ::
 CALL HCHAR(24,31,37)
3ØØ CALL KEY(Ø,KE,S):: IF S=
Ø THEN 3ØØ
31Ø CALL CLEAR
32Ø DISPLAY AT(1,1):"HOCHREC
HNUNGS-WERTE?" :: DISPLAY AT
(2,1):"MAXIMAL 22 WERTE" ::
ACCEPT AT(1,25):M :: CALL HC
HAR(2,3,33,28)
33Ø G=EXP(A)-1 :: WT=EXP(T/N
-A*(N-1)/2)
34Ø CALL HCHAR(1,3,32,28)::
DISPLAY AT(1,2):"HOCHRECHNUN
G DER PERIODEN"
35Ø K=N :: H=(G+1)^K
36Ø FOR I=1 TO M
37Ø WE(I)=WT*H :: WE(I)=INT(
WE(I)*1ØØ+.5)/1ØØ

38Ø DISPLAY AT(I+2,3):"WERT"
 :: DISPLAY AT(I+2,8):USING
"##":I :: DISPLAY AT(I+2,1Ø)
:":" :: DISPLAY AT(I+2,15):U
SING "######":WE(I)
39Ø K=K+1
4ØØ H=(G+1)^K
41Ø IF I=M THEN 43Ø
42Ø NEXT I
43Ø CALL KEY(Ø,KE,S):: IF S=
Ø THEN 43Ø
44Ø CALL CLEAR :: CALL SCREE
N(16)
45Ø DISPLAY AT(24,1):"HOCHRE
CHNUNG:";M;"PERIODEN"
46Ø S1$="7F7F7F7F7F7F7F7F"
47Ø S2$="ØØ7F7F7F7F7F7F7FØØØ
Ø7F7F7F7F7F7FØØØØØØ7F7F7F7F7
FØØØØØØØØ7F7F7F7F"
48Ø S3$="ØØØØØØØØØØ7F7F7FØØØ
ØØØØØØØØ7F7FØØØØØØØØØØØØØØØ7
FØØØØØØØØØØØØØØØØ"
49Ø CALL CHAR(96,S1$):: CALL
 CHAR(97,S2$):: CALL CHAR(1Ø
1,S3$)

5ØØ S4$="F8Ø8Ø8Ø8Ø8Ø8Ø8Ø8ØFØ
ØØØØØØØØØØØØØFFØØØØØØØØØØØØØ
Ø38Ø8Ø8Ø8Ø8Ø8Ø8Ø8" :: CALL C
HAR(112,S4$):: CALL COLOR(11
,13,1)
51Ø S5$="FØØØØØØØØØØØØØØØ1F1
Ø1Ø1Ø1Ø1Ø1Ø1C1Ø1Ø1Ø1Ø1Ø1
Ø" :: CALL CHAR(116,S5$)
52Ø DISPLAY AT(1Ø,1):"IST 1Ø
Ø% SUMME ODER MAXIMUM": :"DE
R DATEN ? (S/M)"
53Ø ACCEPT AT(14,14):MOD$
54Ø IF MOD$<>"S" AND MOD$<>"
M" THEN 53Ø
55Ø FOR I=1Ø TO 14 :: CALL H
CHAR(I,3,32,28):: NEXT I
56Ø CALL COLOR(9,7,1,1Ø,7,1)
57Ø FOR I=Ø TO 1Ø :: DISPLAY
 AT(2*I+2,1):USING "###":(1Ø
-I)*1Ø :: NEXT I
58Ø FOR I=1 TO 1Ø :: DISPLAY
```

```
AT(2*I,4):"p" :: DISPLAY AT
(2*I+1,4):"s"
590 DISPLAY AT(2*I,27):"u" :
: DISPLAY AT(2*I+1,27):"v" :
: NEXT I
600 DISPLAY AT(22,4):"q"
610 FOR I=1 TO 11 :: DISPLAY
 AT(22,2*I+3):"rr" :: NEXT I
620 DISPLAY AT(22,27):"t"
630 DA=WE(1)
640 FOR I=2 TO M
650 DA=MAX(DA,WE(I))
660 NEXT I
670 IF MOD$="S" THEN 740
680 DISPLAY AT(1,27):"MA"
690 FOR I=1 TO M
700 WE(I)=WE(I)/DA*160
710 WE(I)=INT(WE(I)+.5)
720 NEXT I
730 GOTO 760
740 DA=0 :: DISPLAY AT(1,27)
:"SU" :: FOR I=1 TO M :: DA=
DA+WE(I):: NEXT I
750 GOTO 690
760 FOR I=1 TO M
770 HOEHE=INT(WE(I)/8)
780 ZUS=WE(I)-HOEHE*8
790 CALL VCHAR(21-HOEHE,I+6,
104-ZUS,1)
800 CALL VCHAR(22-HOEHE,I+6,
96,HOEHE)
810 NEXT I
820 CALL KEY(0,KE,S):: IF S=
0 THEN 820
830 W$="E N D E"
840 CALL CLEAR
850 CALL STOTT(12,W$)
860 ZE1=11 :: SP1=13 :: ZE2=
13 :: SP2=21
870 CALL RAHMEN(ZE1,SP1,ZE2,
SP2)
880 DISPLAY AT(24,3):"PROGRA
MMIERT VON R.BITTER"
890 FOR I=1 TO 800 :: NEXT I
900 END
910 SUB STOTT(12,W$)
920 Q=INT(32-LEN(W$))/2+1
930 FOR AX=Q TO LEN(W$)+Q-1
:: S$=SEG$(W$,AX-Q+1,1):: FO
R AZ=31 TO AX STEP -1
940 CALL HCHAR(12,AZ+1,32)::
 CALL HCHAR(12,AZ,ASC(S$))::
 NEXT AZ :: NEXT AX
950 SUBEND
960 SUB RAHMEN(ZE1,SP1,ZE2,S
P2)
970 CALL CHAR(128,"000000FFF
F",129,"1818181818181818",13
0,"0000001F1F181818",131,"00
0000F8F8181818",132,"1818181
F1F",133,"181818F8F8")
980 CALL HCHAR(ZE1,SP1,130):
: CALL HCHAR(ZE1,SP2,131)::
CALL HCHAR(ZE2,SP1,132):: CA
LL HCHAR(ZE2,SP2,133)
990 CALL HCHAR(ZE1,SP1+1,128
,SP2-SP1-1)
1000 CALL HCHAR(ZE2,SP1+1,12
8,SP2-SP1-1)
1010 CALL VCHAR(ZE1+1,SP1,12
9,ZE2-ZE1-1):: CALL VCHAR(ZE
1+1,SP2,129,ZE2-ZE1-1)
1020 SUBEND
1030 SUB EINLEITUNG
1040 CALL CLEAR
1050 ZE1=1 :: SP1=2 :: ZE2=2
4 :: SP2=31
1060 CALL RAHMEN(ZE1,SP1,ZE2
,SP2)
1070 DISPLAY AT(2,8):"HOCHRE
CHNUNGEN" :: CALL HCHAR(3,10
,33,14)
1080 DISPLAY AT(4,1):"DIESES
 PROGRAMM ERRECHNET": :"AUF
GRUND VON MAX.22 FEST-": :"S
TEHENDEN WERTEN BIS ZU 22":
:"WERTE IN DIE ZUKUNFT"
1090 DISPLAY AT(12,1):"(PROG
NOSE).VORHER WIRD DIE": :"MI
TTLERE WACHSTUMSRATE,AUF": :
"GRUND DER EINGEGEBENEN WERT
E"
1100 DISPLAY AT(18,1):"ANGEZ
EIGT.JE MEHR WERTE EIN-": :"
GEGEBEN WERDEN UM SO GENAUER
": :"DIE PROGNOSE.
  >>>"
1110 CALL KEY(0,KE,S):: IF S
=0 THEN 1110
1120 FOR I=4 TO 23 :: CALL H
CHAR(I,3,32,28):: NEXT I
1130 DISPLAY AT(4,1):"NACH D
ER AUSGABE DER PROG-": :"NOS
E KANN DURCH DRUECKEN": :"EI
NER TASTE DIE ERRECHNETEN":
:"WERTE AUF EINER BALKEN-"
1140 DISPLAY AT(12,1):"GRAFI
K AUSGEGEBEN WERDEN."
1150 DISPLAY AT(22,26):">>>"
1160 CALL KEY(0,KE,S):: IF S
=0 THEN 1160
1170 SUBEND
```

Fortsetzung von Seite 64

960–1020 Unterprogramm Rahmen
1030–1170 Unterprogramm Einleitung

Anm. der Redaktion: Der Programmierstil deutet darauf hin, daß der Autor noch nicht allzulange im Besitz des Xbasic-Moduls ist.

TI-Basic und Xbasic-Befehle sind bunt durcheinander gemischt. Durch eine korrekte Programmierung in Xbasic wäre

Bitte lesen Sie weiter auf Seite 69

```
10 ! ************************
11 ! *                      *
12 ! *    WIDERSTAENDE      *
13 ! *                      *
14 ! *    Copyright by      *
15 ! *   Ruediger Hussy     *
16 ! *                      *
17 ! *                      *
19 ! * Benoetigte Geraete   *
20 ! *  TI99/4A  Konsole    *
21 ! *     Ext. Basic       *
22 ! *                      *
26 ! *  Speicherbelegung    *
27 ! *     3891 Bytes       *
28 ! *                      *
29 ! ************************
30 !
290 ON BREAK NEXT :: ON WARN
ING NEXT
300 CALL CLEAR :: CALL SCREE
N(2)
310 X=X+1 :: CALL COLOR(X,11
,2):: IF X=14 THEN 330
320 GOTO 310
330 CALL CLEAR
340 PRINT "...W I D E R S T
A N D....."
350 PRINT :: PRINT
360 PRINT " @...BY...R U E D
 I G E R --................
.............H U S S Y.."
370 PRINT :: PRINT :: PRINT
:: PRINT :: PRINT "BITTE TAS
TE DRUECKEN"
380 PRINT :: PRINT :: PRINT
:: PRINT :: PRINT
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 !
510 !
520 CALL CLEAR
530 PRINT "......M E N U E"
540 Y=0
550 PRINT :: Y=Y+1 :: IF Y=4
 THEN 570
560 GOTO 550
570 PRINT "0 FUER PARALELLSC
HALTUNG" :: PRINT :: PRINT "
1 FUER WIDERSTAND >>>>>.....
..CODFARBE" :: PRINT
580 PRINT "2 FUER CODFARBE..
>>>>>>>......WIDERSTAND" ::
PRINT
590 PRINT "3 ERRECHNEN VON S
PANNUNG" :: PRINT
600 PRINT "4 ERRECHNEN VON S
TROM" :: PRINT
610 PRINT "5 ERRECHNEN VON W
IDERSTAND" :: PRINT
620 PRINT "6 PROGRAMMENDE" :
: PRINT
630 INPUT "IHRE WAHL? ":A$ :
: IF A$="1" THEN 1110 :: IF
A$="6" THEN 950 :: IF A$="0"
 THEN 2000
640 IF A$="2" THEN 960 :: IF
 A$="5" THEN 740 :: IF A$="4
" THEN 650 :: IF A$="3" THEN
 850 :: IF A$>"6" THEN 520
650 CALL CLEAR :: PRINT "ERR
ECHNEN VON STROM" :: PRINT :
: PRINT
660 INPUT "SPANNUNG IN VOLT(
V) ":U
670 INPUT "WIDERSTAND IN OHM
 ":R
680 I=U/R :: PRINT :: PRINT
:: PRINT "ES FLIESSEN":I:"AM
PERE(A)"
690 PRINT :: PRINT :: PRINT
:: INPUT "UMRECHNEN IN mA J/
N":K$
700 IF K$="J" THEN 710 :: IF
 K$="N" THEN 720
710 MA=I*1000 :: PRINT "DAS
SIND":MA:"MILLI AMPERE"
720 PRINT :: PRINT :: PRINT
"TASTE DRUECKEN"
730 CALL KEY(0,K,S):: IF S=0
 THEN 730 :: GOTO 520
740 CALL CLEAR :: PRINT :: P
RINT :: PRINT "ERRECHNEN  VO
N WIDERSTAND " :: PRINT :: P
RINT
750 INPUT "SPANNUNG (V) ":C
760 INPUT "STROM (A) ":ER
770 R=C/ER :: PRINT :: PRINT
 :: PRINT "ES SIND":R:"OHM"
780 PRINT :: PRINT :: PRINT
:: INPUT "UMRECHNEN OHM IN K
ILO OHM J/N":L$
790 IF L$="J" THEN 810 :: IF
 L$="N" THEN 820
800 !
810 KO=R/1000 :: PRINT "DAS
SIND":KO:"KILO OHM"
820 PRINT :: PRINT :: PRINT
"BITTE EINE TASTE DRUECKEN"
830 CALL KEY(0,K,S):: IF S=0
 THEN 830 :: GOTO 520
840 !
850 CALL CLEAR :: PRINT :: P
RINT :: PRINT "ERRECHNEN VON
 SPANNUNG" :: PRINT :: PRINT
860 INPUT "WIDERSTAND IN OHM
 ":WI :: INPUT "STROM IN AMP
ERE ":AM
870 REM
880 UI=WI*AM :: PRINT :: PRI
NT :: PRINT "ES IST EINE SPA
NNUNG VON":UI:"VOLT"
```

```
890 PRINT :: PRINT :: INPUT
"UMRECHNEN IN MV J/N ":VB$
900 IF VB$="J" THEN 920 :: I
F VB$="N" THEN 930
910 REM
920 MV=UI*1000 :: PRINT :: P
RINT :: PRINT "DAS SIND":MV:
"MILLI VOLT"
930 PRINT :: PRINT :: PRINT
"BITTE EINE TASTE DRUECKEN"
940 CALL KEY(0,K,S):: IF S=0
 THEN 940 :: GOTO 520
950 END
960 CALL CLEAR :: PRINT "COD
FARBE > WIDERSTANDSW." :: IN
PUT "CODFARBEN: ":AX$ :: PRI
NT :: PRINT :: PRINT "DIE DR
ITTE RINGFARBE SIND NULLEN"
970 IF AX$="SCHWARZ" THEN 10
80 :: IF AX$="BRAUN" THEN 10
70 :: IF AX$="ROT" THEN
1060 :: IF AX$="ORANGE" THEN
 1050 :: IF AX$="GELB" THEN
1040
980 IF AX$="GRUEN" THEN 1030
 :: IF AX$="BLAU" THEN 1020
:: IF AX$="VIOLETT" THEN 101
0 :: IF AX$="GRAU" THEN 1000
 :: IF AX$="WEISS" THEN 990
990 PRINT "9" :: GOTO 1090
1000 PRINT "8" :: GOTO 1090
1010 PRINT "7" :: GOTO 1090
1020 PRINT "6" :: GOTO 1090
1030 PRINT "5" :: GOTO 1090
1040 PRINT "4" :: GOTO 1090
1050 PRINT "3" :: GOTO 1090
1060 PRINT "2" :: GOTO 1090
1070 PRINT "1" :: GOTO 1090
1080 PRINT "0" :: GOTO 1090
1090 PRINT :: PRINT :: INPUT
 "NOCHMAL ? ":KL$
1100 IF KL$="J" THEN 960 ::
IF KL$<>"J" THEN 520
1110 CALL CLEAR :: PRINT "WI
DERSTANDSW.>CODFARBE" :: PRI
NT :: PRINT
1120 PRINT :: PRINT "DER DRI
TTE RINGWERT IST DIE ANZAHL
DER NULLEN" :: PRINT :: PRIN
T
1130 PRINT :: PRINT " EINGAB
E DES RINGWERTES" :: INPUT A
S$
1140 IF AS$="0" THEN 1250 ::
 IF AS$="1" THEN 1240 :: IF
AS$="2" THEN 1230 :: IF AS$=
"3" THEN 1220 :: IF AS$="4"
THEN 1210 :: IF AS$="5" THEN
 1200
1150 IF AS$="6" THEN 1190 ::
 IF AS$="7" THEN 1180 :: IF
AS$="8" THEN 1170 :: IF AS$=
"9" THEN 1160
1160 PRINT "WEISS"
1170 PRINT "GRAU" :: GOTO 12
60
1180 PRINT "VIOLETT" :: GOTO
 1260
1190 PRINT "BLAU" :: GOTO 12
60
1200 PRINT "GRUEN" :: GOTO 1
260
1210 PRINT "GELB" :: GOTO 12
60
1220 PRINT "ORANGE" :: GOTO
1260
1230 PRINT "ROT" :: GOTO 126
0
1240 PRINT "BRAUN" :: GOTO 1
260
1250 PRINT "SCHW." :: GOTO 1
260
1260 PRINT :: PRINT :: INPUT
 "NOCHEINMAL ? ":BN$
1270 IF BN$="J" THEN 1110 ::
 IF BN$<>"J" THEN 520
2000 CALL CLEAR :: PRINT "ER
RECHNEN DES ERSATZWIDER - ST
ANDES VON PARALELLSCHALT- UN
GEN" :: PRINT :: PRINT
2010 INPUT "WIEVIELE WIDERST
AENDE SIND..PARALELL? (2,3,4
,5) ":KL$ :: PRINT :: PRINT
2020 IF KL$="2" THEN 2050 ::
 IF KL$="3" THEN 2070 :: IF
KL$="4" THEN 2100 :: IF KL$=
"5" THEN 2200 :: IF KL$>"5"
THEN 2010 :: IF KL$<"2" THEN
 2010
2030 !
2040 !
2050 INPUT "1.WERT ":Q :: IN
PUT "2.WERT ":W
2060 Z=(Q*W)/(Q+W):: PRINT "
DER ERSATZWERT IST":Z :: PRI
NT "OHM" :: FOR K=1 TO 500 :
: NEXT K :: GOTO 520
2070 INPUT "WERT 1. ":V :: I
NPUT "WERT 2. ":B :: INPUT "
WERT 3. ":N :: M=(1/V)+(1/B)
+(1/N):: L=1/M
2080 PRINT "DER ERSATZWERT I
ST ":L :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUECKEN"
2090 CALL KEY(0,K,S):: IF S=
0 THEN 2090 :: GOTO 520
2100 INPUT "WERT 1. ":S :: I
NPUT "WERT 2. ":Z :: INPUT "
WERT 3. ":X :: INPUT "WERT 4
. ":C
```

```
2110 N=(1/S)+(1/Z)+(1/X)+(1/
C):: L=1/N
2120 PRINT "DER ERSATZWERT I
ST ":L :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUCKEN"
2130 CALL KEY(Ø,K,S)
2140 IF S=Ø THEN 2130
2150 GOTO 520
2160 !
2190 !
2200 INPUT "WERT 1. ":G :: I
NPUT "WERT 2. ":H :: INPUT "
WERT 3. ":J :: INPUT "WERT 4
. ":K :: INPUT "WERT 5. ":L
2210 U=(1/G)+(1/H)+(1/J)+(1/
K)+(1/L):: A=1/U
2220 PRINT "DER ERSATZWERT I
ST ":A :: PRINT "OHM" :: PRI
NT :: PRINT :: PRINT "BITTE
EINE TASTE DRUCKEN'
2230 CALL KEY(Ø,K,S)
2240 IF S=Ø THEN 2230
2250 GOTO 520
```

Fortsetzung von Seite 66

das Programm wesentlich kürzer ausgefallen.

Trotzdem haben wir uns aus zwei Gründen zu einem Abdruck entschlossen!

Zum ersten fällt es aufgrund der vielen TI-Basic-Befehle nicht mehr allzu schwer, das Programm komplett in TI-Basic umzuschreiben und zum anderen bietet das vorliegende Programm doch eine beachtliche Hilfestellung beim Errechnen der benötigten Werte. Die Formeln und Ergebnisse wurden von uns überprüft und können unbedenklich übernommen werden.

Alle Angaben sind korrekt.

H. König



```
10 REM********************
11 REM*                  *
12 REM* SPACE  INVADERS  *
13 REM*  Copyright  by   *
14 REM* Sascha Dierkes   *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*     5139 Bytes   *
25 REM********************
26 REM
180 RANDOMIZE
190 RESTORE
200 CALL CLEAR
210 PRINT "*SPACE INVADERS*"
: :"SCHIESSEN SIE SO VIELE..
....BOMBEN WIE MOEGLICH AB..
....DESTO EHER SIE DIESE"
220 PRINT "TREFFEN  DESTO  M
EHR........PUNKTE GIBT ES.":
 :"BERUEHREN DIE BOMBEN JEDO
CH DEN BODEN HABEN SIE EINE.
..STATION VERLOREN"
230 PRINT "UNTEN WERDEN DIE
ANZAHL DER RESTLICHEN STATIO
NEN SOWIE..IHRE PUNKTE UND A
NGRIFFS-...WELLE DER INV.";
240 PRINT " ANGEZEIGT."
250 CALL KEY(0,K,S)
260 IF S=0 THEN 250
270 CALL CLEAR
280 PRINT "JOYSTICK VERSION"
: : :
290 INPUT "ANZAHL DER ANGREI
FER PRO W. (2-10)":ANZ
300 IF (ANZ<2)+(ANZ>10)+(ANZ
<>INT(ANZ))THEN 290
310 CALL CLEAR
320 FOR I=3 TO 4
330 CALL COLOR(I,4,2)
340 NEXT I
350 FOR I=10 TO 13
360 CALL COLOR(I,1,1)
370 NEXT I
380 CALL COLOR(2,1,1)
390 FOR I=1 TO 50
400 ZE=INT(RND*9)+1
410 SP=INT(RND*30)+1
420 CALL HCHAR(ZE,SP,46,1)
430 NEXT I
440 CALL CHAR(124,"0103070F1
F3F7FFF")
450 CALL CHAR(125,"80C0E0F0F
8FCFEFF")
460 CALL CHAR(111,"182442818
1422418")
470 CALL CHAR(118,"1818183C3
C3C7E7E")
480 CALL CHAR(119,"7E7E7E7E7
```

```
E7E7EFF")
490 CALL CHAR(126,"FFFEFCF8F
0E0C080")
500 CALL CHAR(123,"FF7F3F1F0
F070301")
510 CALL CHAR(122,"FFFFFFFFF
FFFFFFF")
520 FOR I=1 TO 22
530 READ ZE,SP,CH,WH
540 CALL HCHAR(ZE,SP,CH,WH)
550 NEXT I
560 DATA 1,13,111,1,1,16,111
,1,2,13,118,1,2,16,118,1,3,1
1,124,1,3,13,119,1,3,16,119,
1,3,18,125,1
570 DATA 4,12,122,6
580 DATA 5,12,126,1,5,14,123
,1,5,15,126,1,5,17,123,1,6,1
2,125,1,6,14,124,1,6,15,125,
1,6,17,124,1
590 DATA 7,12,122,6
600 DATA 8,11,123,1,8,14,123
,1,8,15,126,1,8,18,126,1
610 XSCH=11
620 SPIEL=0
630 RSCH=5
640 POS1=15
650 CALL SCREEN(2)
660 CALL COLOR(2,12,2)
670 CALL CHAR(100,"183C7EFFF
F0000FF")
680 CALL CHAR(128,"001818181
8181800")
690 CALL CHAR(129,"181818181
8187EFF")
700 CALL CHAR(136,"8181423C3
C3C3C18")
710 CALL CHAR(137,"2424A57E5
A5A7E99")
720 CALL COLOR(13,12,2)
730 CALL COLOR(14,11,2)
740 CALL COLOR(9,14,2)
750 CALL COLOR(10,10,2)
760 CALL COLOR(11,7,2)
770 CALL COLOR(12,6,2)
780 CALL HCHAR(21,1,100,30)
790 CALL HCHAR(20,POS1,129)
800 CALL HCHAR(22,5,RSCH+48)
810 FOR SCH=9 TO 1 STEP -1
820 CALL HCHAR(22,27,SCH+48)
830 SPIEL=SPIEL+1
840 HPOS=XSCH
850 HPOS2=INT(RND*(2*SCH+XSC
H-4))+1
860 CALL HCHAR(HPOS-1,HPOS2,
137)
870 CALL HCHAR(HPOS,HPOS2,13
6)
880 HPOS=HPOS+1
890 IF HPOS=20 THEN 1460
900 FOR I=1 TO SCH
910 CALL KEY(1,K,S)
920 CALL JOYST(1,X,Y)
930 IF (S=0)*(X=0)THEN 970
940 GOTO 1200
950 CALL HCHAR(20,POS1,129)
960 CALL SOUND(-50,440,5)
970 NEXT I
980 CALL HCHAR(HPOS-1,HPOS2,
127)
990 GOTO 870
1000 IF SPIEL/ANZ<>INT(SPIEL
/ANZ)THEN 830
1010 NEXT SCH
1020 RSCH=RSCH+XSCH-10
1030 IF RSCH<10 THEN 1050
1040 RSCH=9
1050 XSCH=XSCH+1
1060 CALL HCHAR(22,5,RSCH+48
)
1070 PUNKT=PUNKT+100*XSCH
1080 CALL SOUND(300,524,1)
1090 CALL SOUND(300,588,1)
1100 CALL SOUND(600,660,1)
1110 PUNKT=PUNKT+100*XSCH
1120 CALL SOUND(225,698,3,88
0,1)
1130 CALL SOUND(75,784,1)
1140 CALL SOUND(150,880,1)
1150 CALL SOUND(150,988,2,78
2,1)
1160 CALL SOUND(300,1046,5,7
82,4,658,0)
1170 IF XSCH>16 THEN 1690
1180 CALL COLOR(14,XSCH,2)
1190 GOTO 810
1200 IF X<>-4 THEN 1250
1210 IF POS1-1<1 THEN 1240
1220 CALL HCHAR(20,POS1,127)
1230 POS1=POS1-1
1240 GOTO 950
1250 IF X<>4 THEN 1300
1260 IF POS1+1>30 THEN 1290
1270 CALL HCHAR(20,POS1,127)
1280 POS1=POS1+1
1290 GOTO 950
1300 IF K<>18 THEN 970
1310 FOR J=19 TO HPOS STEP -
1
1320 CALL HCHAR(J,POS1,128)
1330 NEXT J
1340 IF HPOS2<>POS1 THEN 143
0
1350 CALL SOUND(-200,1200,0)
1360 PUNKT=PUNKT+(21-HPOS)*(
11-SCH)
1370 HILF$=STR$(PUNKT)
1380 FOR J=1 TO LEN(HILF$)
1390 CALL HCHAR(22,15+J,ASC(
SEG$(HILF$,J,1)))
1400 NEXT J
```

```
1410 CALL VCHAR(10,POS1,127,
10)
1420 GOTO 1000
1430 CALL VCHAR(HPOS,POS1,12
7,20-HPOS)
1440 CALL SOUND(-200,-2,0)
1450 GOTO 950
1460 RSCH=RSCH-1
1470 CALL SCREEN(10)
1480 CALL SOUND(-1000,-6,0)
1490 CALL HCHAR(10,1,127,352
)
1500 CALL SCREEN(2)
1510 IF RSCH<0 THEN 1550
1520 CALL HCHAR(22,5,RSCH+48
)
1530 CALL HCHAR(20,POS1,129)
1540 GOTO 1000
1550 CALL CLEAR
1560 FOR I=5 TO 8
1570 CALL COLOR(I,16,2)
1580 NEXT I
1590 PRINT "SIE HABEN LEIDER
 VERLOREN":"ABER PROBIEREN S
IE ES DOCH  RUHIG NOCH EINMA
L."
1600 PRINT "SIE HATTEN ";PUN
KT;" PUNKTE":"UND HATTEN SCH
WIERIGKEITS-  GRAD :";SCH
1610 PRINT "IN STUFE:";XSCH-
11
1620 PRINT "NOCHMAL?"
1630 CALL KEY(0,K,S)
1640 IF S=0 THEN 1630
1650 IF K<>74 THEN 1670
1660 GOTO 100
1670 IF K<>78 THEN 1630
1680 END
1690 CALL CLEAR
1700 FOR I=1 TO 14
1710 CALL COLOR(I,2,16)
1720 NEXT I
1730 CALL SCREEN(16)
1740 PRINT "BRAVO!!": :"ABSO
LUTE SPITZENKLASSE"
1750 PRINT "PUNKTE:";PUNKT:"
SIE HABEN ES BIS ZUM LETZTEN
SCHWIERIGKEITSGRAD DURCHGE-
HALTEN!"
1760 PRINT "NOCHMAL?"
1770 CALL KEY(0,K,S)
1780 IF S=0 THEN 1770
1790 IF K=74 THEN 100
1800 IF K<>78 THEN 1770
1810 END
```



```
1Ø ! ***********************
11 ! *                     *
12 ! * ROUTINE ZUM AUS-    *
13 ! *   DRUCKEN VON       *
14 ! * DFUE-PROTOKOLLEN    *
15 ! *   Copyright by      *
16 ! *    Dieter Orth      *
17 ! *                     *
19 ! * Benoetigte Geraete  *
2Ø ! *  TI99/4A  Konsole   *
21 ! *     Ext. Basic      *
22 ! *  Terminal-Emulator  *
23 ! *        oder         *
24 ! *TE II und Disk-Drive *
25 ! *      Drucker        *
26 ! *  Speicherbelegung   *
27 ! *     2754 Bytes      *
28 ! *                     *
29 ! ***********************

1ØØ CALL CLEAR :: CALL SCREE
N(2)
11Ø DISPLAY AT(1,1):"*******
***********************" :: D
ISPLAY AT(2,1):"*" :: DISPLA
Y AT(2,28):"*"
12Ø DISPLAY AT(3,1):"*......
.LESEPROGRAMM.......*" :: DI
SPLAY AT(4,1):"*" :: DISPLAY
 AT(4,28):"*"
13Ø DISPLAY AT(5,1):"*...FUE
R DFUE-TEXTE VON....*" :: DI
SPLAY AT(6,1):"*" :: DISPLAY
 AT(6,28):"*"
14Ø DISPLAY AT(7,1):"*......
.'DISKETTEN'.........*" :: DI
SPLAY AT(8,1):"*" :: DISPLAY
 AT(8,28):"*"
15Ø DISPLAY AT(9,1):"*......
..IM FORMAT.........*" :: DI
SPLAY AT(1Ø,1):"*" :: DISPLA
Y AT(1Ø,28):"*" :: DISPLAY A
T(11,1):"*  DISPLAY/VARIABLE
 8Ø/128 *" :: DISPLAY AT(11,
28):"*"
16Ø DISPLAY AT(12,1):"*" ::
DISPLAY AT(12,28):"*" :: DIS
PLAY AT(13,1):"*" :: DISPLAY
 AT(13,28):"*" :: DISPLAY AT
(14,1):"*" :: DISPLAY AT(14,
28):"*"
17Ø DISPLAY AT(15,1):"*.....
...VERSION 1.2 ......*" :: D
ISPLAY AT(16,1):"*" :: DISPL
AY AT(16,28):"*" :: DISPLAY
AT(17,1):"******************
**********"
18Ø CALL SCREEN(4):: FOR I=1
 TO 75Ø :: NEXT I
```

```
190 !----------------------
200 ! ROUTINE ZUM AUSDRUCKEN
      DER DFUE-PROTOKOLLE
      VON DISKETTEN
210 ! LITERATUR:
      TI REVUE 2/84,S.44
220 ! CR BY DIETER ORTH
          HARTGERSTR. 12
       33 BRAUNSCHWEIG
230 !----------------------
     MENUE
     ----------------------
240 CALL CLEAR :: DISPLAY AT
(2,4):"<GERAETE.DATEINAMEN>"
250 !----------------------
260 DISPLAY AT(4,4):"===>" :
: ACCEPT AT(4,8)SIZE(15)VALI
DATE(UALPHA,DIGIT,"./-_"):FI
LE$ :: IF FILE$="" THEN 260
270 !----------------------
280 DISPLAY AT(7,4):"<TERMIN
AL-EMULATOR>" :: DISPLAY AT(
8,4):"------------------"
290 DISPLAY AT(9,4):"TE II
 =<1>" :: DISPLAY AT(10,17):
"> <" :: DISPLAY AT(11,4):"T
E 1200 =<2>" :: ACCEPT AT(10
,18)SIZE(1)VALIDATE("12"):T$
300 IF T$="" THEN 290 :: IF
T$="1" THEN 320 :: IF T$="2"
 THEN 330 ELSE 290
310 DISPLAY AT(24,1):FILE$
320 OPEN #1:FILE$,INPUT ,DIS
PLAY ,VARIABLE 80 :: GOTO 34
0
330 OPEN #1:FILE$,INPUT ,DIS
PLAY ,VARIABLE 128
340 DISPLAY AT(14,4):"DRUCKE
R  J/N > <" :: ACCEPT AT(14,
18)SIZE(1)VALIDATE("JN "):C$
350 IF C$="" THEN 340 :: IF
C$="J" THEN 380 :: IF C$="N"
 THEN 500
360 DISPLAY AT(24,1):FILE$
370 !----------------------
     DRUCKER-AUSGABE
     ----------------------
380 DISPLAY AT(17,4):"RS232.
...<1>" :: DISPLAY AT(18,17)
:"> <" :: DISPLAY AT(19,4):"
SELEKTIV <2>" :: ACCEPT AT(1
8,18)SIZE(1)VALIDATE("12"):D
$
390 IF D$="" THEN 380 :: IF
D$="1" THEN 400 :: IF D$="2"
 THEN 420
400 OPEN #2:"RS232.BA=9600.D
A=8.PA=E",OUTPUT,DISPLAY ::
GOTO 450
410 !----------------------
420 DISPLAY AT(21,4):"DRUCKE
RNAME:" :: ACCEPT AT(22,4)SI
ZE(28)VALIDATE(UALPHA,DIGIT,
".=,"):P$ :: IF P$="" THEN 4
20
430 OPEN #2:P$,OUTPUT,DISPLA
Y
440 !----------------------
450 LINPUT #1:A$
460 IF EOF(1)THEN 570
470 PRINT #2:A$
480 GOTO 450
490 !----------------------
     BILDSCHIRM-AUSGABE
     ----------------------
500 CALL CLEAR :: DISPLAY AT
(21,1):"BITTE DIE SPACE - TA
STE.....BENUTZEN!" :: FOR I=
1 TO 750 :: NEXT I :: CALL C
LEAR
510 LINPUT #1:A$
520 IF EOF(1)THEN 570
530 PRINT A$ :: FOR I=0 TO 1
540 CALL KEY(0,K,S):: IF S=0
 THEN 540 :: IF K=32 THEN 55
0 :: GOTO 540
550 NEXT I :: GOTO 510
560 !----------------------
     MENUE
     ----------------------
570 DISPLAY AT(24,4):"WEITER
...J/N > <" :: ACCEPT AT(24,
18)SIZE(1)VALIDATE("JN"):S$
:: IF S$="" THEN 570 :: IF S
$="J" THEN RUN 240 :: IF S$=
"N" THEN 580
580 RUN "DSK1.LOAD" :: REM--
--->RUN "DSK1.MENUE"
590 END
```

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE



KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre TI REVUE nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Euer" Kiosk nicht beliefert wurde? Kein Problem! Für ganze 30 DM liefern wir Euch per Post sechs Hefte ins Haus (Ausland 40 DM). Einfach den Bestellschein auf der nächsten Seite ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). TI REVUE kommt dann pünktlich ins Haus.

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen– Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für sechs Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

DAS SUPER-SONDER-ANGEBOT: PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen ab sofort die TI-Revue: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater Inserenten, keine gewerblichen Anzeigen. Die kosten pro Millimeter DM 2,50 plus Mehrwertsteuer!

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes – Computer – gehören, nicht abzudrucken oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeitenteils zuläßt.

KASSETTEN

Selbstverständlich gibt es in diesem Heft wieder einen Kassetten-Service. Dabei können wir Ihnen allerdings die Qual der Wahl nicht ersparen. Denn: In Ihrem Interesse haben wir dieses Special sozusagen „geviertelt".
Viertel 1: Alle Utilities und Anwender-Programme in TI Basic
Viertel 2: Alle Spiele in TI Basic
Viertel 3: Alle Utilities/Anwenderprogramme in Extended Basic
Viertel 4: Alle Spiele in Extended Basic.
Und weil es uns Spaß macht, haben wir noch ein fünftes Viertel angehängt: den Disketten-Service. Da sind dann allerdings alle Programme drauf. Und nun zu den harten, nackten Geschäftsbedingungen.
Sie können zahlen:
Gegen Bankabbuchung am Versandtag
Gegen Vorausrechnung, wobei am Tag des Geldeinganges der Versand erfolgt.
Sie können kombinieren:
Jedwedes Programmpaket miteinander — etwa 1 + 4 oder 1,2,3 oder nur 4 und so weiter.

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatenbieter: maximal acht Zeilen à 28 Anschläge.

TI REVUE
Anzeigenabteilg.
Postfach 1107
8044 Lohhof

Name ____________
Vorname ____________
Str.Nr. ____________
(PLZ) Ort ____________

ABO SERVICE-KARTE

6
TI

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten sechs Ausgaben an untenstehende Anschrift.

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

Name ____________
Vorname ____________
Straße/Hausnr. ____________
Plz/Ort ____________
Ich bezahle:
☐ per beiliegendem Verrechnungsscheck
☐ gegen Rechnung
☐ bargeldlos per Bankeinzug von meinem Konto
bei (Bank) und Ort ____________
Kontonummer ____________
Bankleitzahl ____________
(steht auf jedem Kontoauszug)
Unterschrift ____________

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift ____________

TI REVUE

Abo-Service SP2
Postfach 1107
8044 UNTERSCHLEISSHEIM

SERVICE

Das sind die Preise: Pro einzelnes Programm-Paket DM 20,–
Für zwei Programmpakete DM 35,– (5,– DM Ersparnis)
Für drei Programmpakete DM 50,– (10,– DM Ersparnis)
Für alle vier Programmpakete DM 60,– (20,– DM Ersparnis)
(Jedes Programmpaket auf Kassette!)
Für die komplette Diskette – alle vier Programm-Pakete – DM 75,– (hier ist keine Zusammenstellung möglich!)

Die Inhalte der einzelnen Programm-Pakete ersehen Sie aus nachfolgender Übersicht, die Komplett-Diskette enthält alle vier Programm-Pakete!

ACHTUNG: KEIN GELD EINSENDEN! NUR DEN BERECHTIGUNGSCOUPON! WIR MELDEN UNS, WENN DIE WARE VERSANDBEREIT IST!!! SELBSTVERSTÄNDLICH VOLLES UMTAUSCHRECHT BEI NICHTFUNKTIONIEREN.

Programm-Paket 1:
Alle Spiele in TI-Basic
wie Erdbeer-Paul, Apfelfänger, Seeschlacht, Space Invaders, Castle of Dom, Catch with Car, Sky-Driver, TI-Mühle, Irvel.

Programmpaket 2:
Alle Spiele in Ext.-Basic
wie Space 2000, Tore schießen, TI-Fall, Spickzettel, Adventage, Ecarte, Boogabo.

Programmpaket 3:
Alle Anwenderprogramme, Utilities usw. in Basic
wie Dreiecks- und Kreisberechnung, Kompositionen, Dissassembler, Copyright-Zeichen.

Programmpaket 4:
Alle Anwenderprogramme, Utilities usw. in Ext. Basic wie Hexdump, Supergrafik, Toch Typing, Sprite-Set DFÜ, Tabellen-Erstellung, Steuerberechnung, Prognose, Karnoug Veitch, Reset, 3x3-Matrix, Softplot oder Pokelisten.

ACHTUNG! Wir weisen ausdrücklich darauf hin, daß das Spiel „Irvel" wegen Überlänge nur in der Ursprungsversion des Autors kopiert wurde. Eventuell müssen Sie daher dieses Programm gemäß unserem Listing überprüfen.

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen
Programmsammlung 1 ( ) **Programmsammlung 2 ( )**
Programmsammlung 3 ( ) **Programmsammlung 4 ( )**
die Komplett-Diskette ( )
Zutreffendes bitte ankreuzen!

Ich zahle:

Gegen Vorausrechnung ( ) – Versand am Tag des Geldeinganges
Gegen Bankabbuchung am Versandtag ( )
Meine Bank (mit Ortsname) ....
Meine Kontonummer ....
Meine Bankleitzahl .... (steht auf jedem Bankauszug)

Vorname .... Nachname ....

Str./Nr. .... PLZ / Ort ....

Hiermit bestätigen ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift ....

Bitte ausschneiden und einsenden an
TI-SPECIAL 2
KASSETTENSERVICE
Postfach 1107
8044 Unterschleißheim

```
1Ø ! *********************
11 ! *                   *
12 ! *   KARNOUGH-VEITCH *
13 ! *      DIAGRAMM     *
14 ! *    Copyright by   *
15 ! *                   *
16 ! *   Bernard Schmidt *
17 ! *                   *
19 ! * Benoetigte Geraete *
2Ø ! * `TI99/4A  Konsole *
21 ! *     Ext. Basic    *
22 ! *                   *
26 ! *  Speicherbelegung *
27 ! *     11702 Bytes   *
28 ! *                   *
29 ! *********************
3Ø !
26Ø CALL CLEAR
27Ø DIM C(16),R$(62)
28Ø !
29Ø ! ZEICHENDEFINITIONEN
3ØØ DATA "7CØØØØ38447C4444",
"7CØØØØ7824382478","7CØØØØ3C
4Ø4Ø4Ø3C","7CØØØØ7824242478"
31Ø DATA "Ø1Ø1Ø1Ø1Ø1Ø1Ø1Ø1",
"ØØØØØØØØØØØØØØFF","Ø1Ø1Ø1Ø1
Ø1Ø1Ø1FF","FFØØØØØØØØØØØØØØ"
,"3Ø8Ø8Ø8Ø8Ø8Ø3Ø8Ø"
32Ø RESTORE 3ØØ :: FOR I=1 T
O 4 :: READ A$ :: CALL CHAR(
1ØØ+I,A$):: NEXT I
33Ø RESTORE 31Ø :: FOR I=1 T
O 5 :: READ A$ :: CALL CHAR(
32+I,A$):: NEXT I
34Ø !
35Ø !DATEN FUER WAHRHEITSTAB
ELLE
36Ø !FUER Ø
37Ø DATA 5,5,5,7,5,9,5,11,6,
5,6,7,6,9,7,5,7,7,7,11,8,5,8
,7,9,5,9,9,9,11
38Ø DATA 1Ø,5,1Ø,9,11,5,11,1
1,12,5,13,7,13,9,13,11,14,7,
14,9,15,7,15,11,16,7,17,9,17
,11
39Ø DATA 18,9,19,11
4ØØ !FUER 1
41Ø DATA 6,11,7,9,8,9,8,11,9
,7,1Ø,7,1Ø,11,11,7,11,9,12,7
,12,9,12,11,13,5,14,5,14,11
42Ø DATA 15,5,15,9,16,5,16,9
,16,11,17,5,17,7,18,5,18,7,1
8,11,19,5,19,7,19,9,2Ø,5,2Ø,
7,2Ø,9,2Ø,11
43Ø !
44Ø !ERGEBNISDATEN
45Ø DATA Ø,1,f,d,b,e,c,a,h,g
,ef,ed,eb,eh,fc,cd,bc,ch,af,
ad,ab,ah,fg,gd,bg,gh,eg
46Ø DATA ec,ac,ag,fh,fd,bd,b
h,efg,egd,ebg,efc,ecd,ebc,af
c,acd,abc,afg,agd,abg,egh
47Ø DATA ech,ach,agh,efh,efd
,ebd,ebh,fch,fcd,bcd,bch,afh
,afd,abd,abh,fgh,fgd,bgd,bgh
48Ø DATA efgh,efgd,efch,efcd
,ebgh,ebgd,ebch,ebcd,afgh,af
gd,afch,afcd,abgh,abgd,abch,
abcd
49Ø !DARSTELLUNG WAHRHEITSTA
BELLE
5ØØ A$="WAHRHEITSTABELLE" ::
 FOR I=1 TO 16 :: CALL HCHAR
(1,2+I,ASC(SEG$(A$,I,1))))::
NEXT I
51Ø FOR I=4 TO 12 STEP 2 ::
CALL HCHAR(4,I,35):: CALL HC
HAR(3,I,33):: NEXT I
52Ø FOR I=3 TO 13 STEP 2 ::
CALL HCHAR(4,I,34):: NEXT I
53Ø FOR I=4 TO 12 STEP 2 ::
FOR J=5 TO 2Ø :: CALL HCHAR(
J,I,33):: NEXT J :: NEXT I
54Ø FOR I=1 TO 4 :: J=2*I+3
:: CALL HCHAR(3,J,96+I):: NE
XT I
55Ø CALL HCHAR(3,13,65)
56Ø FOR I=1 TO 1Ø :: CALL HC
HAR(4+I,3,47+I):: NEXT I
57Ø FOR I=1 TO 6 :: CALL HCH
AR(14+I,2,49):: CALL HCHAR(1
4+I,3,47+I):: NEXT I
58Ø RESTORE 37Ø :: FOR I=1 T
O 32 :: READ A,B :: CALL HCH
AR(A,B,48):: NEXT I
59Ø RESTORE 41Ø :: FOR I=1 T
O 32 :: READ A,B :: CALL HCH
AR(A,B,49):: NEXT I
6ØØ A$="KORREKTUR  ->K" :: F
OR I=1 TO 14 :: CALL HCHAR(3
,17+I,ASC(SEG$(A$,I,1))):: N
EXT I
61Ø A$="1=HIGH" :: FOR I=1 T
O 6 :: CALL HCHAR(6,17+I,ASC
(SEG$(A$,I,1))):: NEXT I
62Ø A$="Ø=LOW" :: FOR I=1 TO
 5 :: CALL HCHAR(8,17+I,ASC(
SEG$(A$,I,1))):: NEXT I
63Ø A$="X=DON'T CARE" :: FOR
 I=1 TO 12 :: CALL HCHAR(1Ø,
17+I,ASC(SEG$(A$,I,1))):: NE
XT I
64Ø A$="BITTE A EINGEBEN :"
:: FOR I=1 TO 18 :: CALL HCH
AR(22,3+I,ASC(SEG$(A$,I,1)))
:: NEXT I
65Ø !
66Ø !WERTE FUER A EINGEBEN
67Ø FOR I=Ø TO 15
68Ø CALL KEY(Ø,K,S):: IF S<=
Ø THEN 68Ø :: IF K=48 OR K=4
```

```
9 OR K=88 THEN 690 ELSE IF K
=75 THEN 700 ELSE 680
690 CALL HCHAR(5+I,13,K):: G
OTO 710
700 IF I=0 THEN GOTO 680 ELS
E I=I-1 :: GOTO 680
710 NEXT I
720 CALL HCHAR(22,3,32,19)::
 CALL HCHAR(3,17,32,15):: FO
R I=1 TO 5 STEP 2 :: CALL HC
HAR(5+I,17,32,13):: NEXT I
730 !
740 !KV-DIAGRAMM ERSTELLEN
750 A$="KV-DIAGRAMM" :: FOR
I=1 TO 11 :: CALL HCHAR(3,17
+I,ASC(SEG$(A$,I,1))):: NEXT
 I
760 CALL HCHAR(7,20,34,4)::
CALL VCHAR(8,19,33,4):: CALL
 VCHAR(8,24,37,4):: CALL HCH
AR(12,20,36,4)
770 CALL HCHAR(6,21,36,2)::
CALL HCHAR(5,21,100):: CALL
VCHAR(9,25,33,2):: CALL VCHA
R(9,26,99)
780 CALL VCHAR(10,18,37,2)::
 CALL VCHAR(10,17,97):: CALL
 HCHAR(13,22,34,2):: CALL HC
HAR(14,22,98)
790 FOR I=0 TO 15 :: CALL GC
HAR(5+I,13,C(I)):: NEXT I
800 CALL HCHAR(8,20,C(0))::
CALL HCHAR(8,21,C(1)):: CALL
 HCHAR(9,20,C(2)):: CALL HCH
AR(9,21,C(3))
810 CALL HCHAR(8,23,C(4))::
CALL HCHAR(8,22,C(5)):: CALL
 HCHAR(9,23,C(6)):: CALL HCH
AR(9,22,C(7))
820 CALL HCHAR(11,20,C(8))::
 CALL HCHAR(11,21,C(9)):: CA
LL HCHAR(10,20,C(10)):: CALL
 HCHAR(10,21,C(11))
830 CALL HCHAR(11,23,C(12)):
: CALL HCHAR(11,22,C(13))::
CALL HCHAR(10,23,C(14)):: CA
LL HCHAR(10,22,C(15))
840 !
850 !AUSWERTUNG
860 FOR I=0 TO 15 :: C(I)=C(
I)-48 :: NEXT I
870 RESTORE 450
880 FOR I=0 TO 81 :: READ R$
(I):: NEXT I :: ERG$=""
890 !
900 !16er BLOCK
910 A=0 :: FOR I=0 TO 15 ::
A=A+C(I):: NEXT I
920 IF A=0 THEN ERG$=R$(0)EL
SE 930 :: GOTO 2350
930 FOR I=0 TO 15 :: IF C(I)
=0 THEN 960
940 NEXT I :: ERG$=R$(1):: G
OTO 2350
950 !
960 !8er BLOCK
970 IF C(0)>=1 AND C(1)>=1 A
ND C(2)>=1 AND C(3)>=1 AND C
(10)>=1 AND C(11)>=1 AND C(8
)>=1 AND C(9)>=1 THEN ERG$=E
RG$&R$(2)&"+" ELSE 990
980 FOR I=0 TO 3 :: FOR J=8
TO 11 :: C(I),C(J)=0 :: NEXT
 J :: NEXT I
990 IF C(1)>=1 AND C(5)>=1 A
ND C(3)>=1 AND C(7)>=1 AND C
(11)>=1 AND C(15)>=1 AND C(9
)>=1 AND C(13)>=1 THEN ERG$=
ERG$&R$(3)&"+" ELSE 1010
1000 C(1),C(5),C(3),C(7),C(1
1),C(15),C(9),C(13)=0
1010 IF C(5)>=1 AND C(7)>=1
AND C(15)>=1 AND C(13)>=1 AN
D C(4)>=1 AND C(6)>=1 AND C(
14)>=1 AND C(12)>=1 THEN ERG
$=ERG$&R$(4)&"+" ELSE 1030
1020 C(5),C(7),C(15),C(13),C
(4),C(6),C(14),C(12)=0
1030 IF C(0)>=1 AND C(1)>=1
AND C(2)>=1 AND C(3)>=1 AND
C(5)>=1 AND C(6)>=1 AND C(7)
>=1 AND C(4)>=1 THEN ERG$=ER
G$&R$(5)&"+" ELSE 1050
1040 FOR I=0 TO 3 :: FOR J=4
 TO 7 :: C(I),C(J)=0 :: NEXT
 J :: NEXT I
1050 IF C(2)>=1 AND C(3)>=1
AND C(7)>=1 AND C(6)>=1 AND
C(10)>=1 AND C(11)>=1 AND C(
15)>=1 AND C(14)>=1 THEN ERG
$=ERG$&R$(6)&"+" ELSE 1070
1060 C(2),C(3),C(7),C(6),C(1
0),C(11),C(15),C(14)=0
1070 IF C(8)>=1 AND C(9)>=1
AND C(10)>=1 AND C(11)>=1 AN
D C(12)>=1 AND C(13)>=1 AND
C(14)>=1 AND C(15)>=1 THEN E
RG$=ERG$&R$(7)&"+" ELSE 1090
1080 FOR I=8 TO 15 :: C(I)=0
 :: NEXT I
1090 IF C(0)>=1 AND C(2)>=1
AND C(4)>=1 AND C(6)>=1 AND
C(8)>=1 AND C(10)>=1 AND C(1
2)>=1 AND C(14)>=1 THEN ERG$
=ERG$&R$(8)&"+" ELSE 1110
1100 FOR I=0 TO 14 STEP 2 ::
 C(I)=0 :: NEXT I
1110 IF C(0)>=1 AND C(1)>=1
AND C(5)>=1 AND C(4)>=1 AND
C(8)>=1 AND C(9)>=1 AND C(13
)>=1 AND C(12)>=1 THEN ERG$=
```

```
ERG$&R$(9)&"+" ELSE 114Ø
112Ø C(Ø),C(1),C(5),C(4),C(8
),C(9),C(13),C(12)=Ø
113Ø !
114Ø !4er BLOCK
115Ø IF C(Ø)>=1 AND C(1)>=1
AND C(2)>=1 AND C(3)>=1 THEN
 ERG$=ERG$&R$(1Ø)&"+" ELSE 1
17Ø
116Ø FOR I=Ø TO 3 :: C(I)=Ø
:: NEXT I
117Ø IF C(1)>=1 AND C(3)>=1
AND C(5)>=1 AND C(7)>=1 THEN
 ERG$=ERG$&R$(11)&"+" ELSE 1
19Ø
118Ø FOR I=1 TO 7 STEP 2 ::
C(I)=Ø :: NEXT I
119Ø IF C(4)>=1 AND C(5)>=1
AND C(6)>=1 AND C(7)>=1 THEN
 ERG$=ERG$&R$(12)&"+" ELSE 1
21Ø
12ØØ FOR I=4 TO 7 :: C(I)=Ø
:: NEXT I
121Ø IF C(Ø)>=1 AND C(2)>=1
AND C(4)>=1 AND C(6)>=1 THEN
 ERG$=ERG$&R$(13)&"+" ELSE 1
23Ø
122Ø FOR I=Ø TO 6 STEP 2 ::
C(I)=Ø :: NEXT I
123Ø IF C(2)>=1 AND C(3)>=1
AND C(1Ø)>=1 AND C(11)>=1 TH
EN ERG$=ERG$&R$(14)&"+" ELSE
 125Ø
124Ø C(2),C(3),C(1Ø),C(11)=Ø
125Ø IF C(3)>=1 AND C(7)>=1
AND C(11)>=1 AND C(15)>=1 TH
EN ERG$=ERG$&R$(15)&"+" ELSE
 127Ø
126Ø C(3),C(7),C(11),C(15)=Ø
127Ø IF C(7)>=1 AND C(6)>=1
AND C(15)>=1 AND C(14)>=1 TH
EN ERG$=ERG$&R$(16)&"+" ELSE
 129Ø
128Ø C(7),C(6),C(15),C(14)=Ø
129Ø IF C(2)>=1 AND C(1Ø)>=1
 AND C(6)>=1 AND C(14)>=1 TH
EN ERG$=ERG$&R$(17)&"+" ELSE
 131Ø
13ØØ C(2),C(1Ø),C(6),C(14)=Ø
131Ø IF C(8)>=1 AND C(9)>=1
AND C(1Ø)>=1 AND C(11)>=1 TH
EN ERG$=ERG$&R$(18)&"+" ELSE
 133Ø
132Ø FOR I=8 TO 11 :: C(I)=Ø
 :: NEXT I
133Ø IF C(9)>=1 AND C(11)>=1
 AND C(13)>=1 AND C(15)>=1 T
HEN ERG$=ERG$&R$(19)&"+" ELS
E 135Ø
134Ø FOR I=9 TO 15 STEP 2 ::
 C(I)=Ø :: NEXT I
135Ø IF C(12)>=1 AND C(13)>=
1 AND C(14)>=1 AND C(15)>=1
THEN ERG$=ERG$&R$(2Ø)&"+" EL
SE 137Ø
136Ø FOR I=12 TO 15 :: C(I)=
Ø :: NEXT I
137Ø IF C(8)>=1 AND C(1Ø)>=1
 AND C(12)>=1 AND C(14)>=1 T
HEN ERG$=ERG$&R$(21)&"+" ELS
E 139Ø
138Ø FOR I=8 TO 14 STEP 2 ::
 C(I)=Ø :: NEXT I
139Ø IF C(Ø)>=1 AND C(1)>=1
AND C(8)>=1 AND C(9)>=1 THEN
 ERG$=ERG$&R$(22)&"+" ELSE 1
41Ø
14ØØ C(Ø),C(1),C(8),C(9)=Ø
141Ø IF C(1)>=1 AND C(5)>=1
AND C(9)>=1 AND C(13)>=1 THE
N ERG$=ERG$&R$(23)&"+" ELSE
143Ø
142Ø C(1),C(5),C(9),C(13)=Ø
143Ø IF C(5)>=1 AND C(4)>=1
AND C(13)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(24)&"+" ELSE
 145Ø
144Ø C(5),C(4),C(13),C(12)=Ø
145Ø IF C(Ø)>=1 AND C(4)>=1
AND C(8)>=1 AND C(12)>=1 THE
N ERG$=ERG$&R$(25)&"+" ELSE
147Ø
146Ø C(Ø),C(4),C(8),C(12)=Ø
147Ø IF C(Ø)>=1 AND C(1)>=1
AND C(5)>=1 AND C(4)>=1 THEN
 ERG$=ERG$&R$(26)&"+" ELSE 1
49Ø
148Ø C(Ø),C(1),C(5),C(4)=Ø
149Ø IF C(2)>=1 AND C(3)>=1
AND C(7)>=1 AND C(6)>=1 THEN
 ERG$=ERG$&R$(27)&"+" ELSE 1
51Ø
15ØØ C(2),C(3),C(7),C(6)=Ø
151Ø IF C(1Ø)>=1 AND C(11)>=
1 AND C(15)>=1 AND C(14)>=1
THEN ERG$=ERG$&R$(28)&"+" EL
SE 153Ø
152Ø C(1Ø),C(11),C(15),C(14)
=Ø
153Ø IF C(8)>=1 AND C(9)>=1
AND C(13)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(29)&"+" ELSE
 155Ø
154Ø C(8),C(9),C(13),C(12)=Ø
155Ø IF C(Ø)>=1 AND C(2)>=1
AND C(1Ø)>=1 AND C(8)>=1 THE
N ERG$=ERG$&R$(3Ø)&"+" ELSE
157Ø
156Ø C(Ø),C(2),C(1Ø),C(8)=Ø
157Ø IF C(1)>=1 AND C(3)>=1
AND C(11)>=1 AND C(9)>=1 THE
```

```
N ERG$=ERG$&R$(31)&"+" ELSE
 159Ø
158Ø C(1),C(3),C(11),C(9)=Ø
159Ø IF C(5)>=1 AND C(7)>=1
AND C(15)>=1 AND C(13)>=1 TH
EN ERG$=ERG$&R$(32)&"+" ELSE
 161Ø
16ØØ C(5),C(7),C(15),C(13)=Ø
161Ø IF C(4)>=1 AND C(6)>=1
AND C(14)>=1 AND C(12)>=1 TH
EN ERG$=ERG$&R$(33)&"+" ELSE
 164Ø
162Ø C(4),C(6),C(14),C(12)=Ø
163Ø !
164Ø !2er BLOCK
165Ø IF C(Ø)>=1 AND C(1)>=1
THEN ERG$=ERG$&R$(34)&"+" EL
SE 167Ø
166Ø C(Ø),C(1)=Ø
167Ø IF C(1)>=1 AND C(5)>=1
THEN ERG$=ERG$&R$(35)&"+" EL
SE 169Ø
168Ø C(1),C(5)=Ø
169Ø IF C(5)>=1 AND C(4)>=1
THEN ERG$=ERG$&R$(36)&"+" EL
SE 171Ø
17ØØ C(5),C(4)=Ø
171Ø IF C(2)>=1 AND C(3)>=1
THEN ERG$=ERG$&R$(37)&"+" EL
SE 173Ø
172Ø C(2),C(3)=Ø
173Ø IF C(3)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(38)&"+" EL
SE 175Ø
174Ø C(3),C(7)=Ø
175Ø IF C(6)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(39)&"+" EL
SE 177Ø
176Ø C(6),C(7)=Ø
177Ø IF C(1Ø)>=1 AND C(11)>=
1 THEN ERG$=ERG$&R$(4Ø)&"+"
ELSE 179Ø
178Ø C(1Ø),C(11)=Ø
179Ø IF C(11)>=1 AND C(15)>=
1 THEN ERG$=ERG$&R$(41)&"+"
ELSE 181Ø
18ØØ C(11),C(15)=Ø
181Ø IF C(15)>=1 AND C(14)>=
1 THEN ERG$=ERG$&R$(42)&"+"
ELSE 183Ø
182Ø C(15),C(14)=Ø
183Ø IF C(8)>=1 AND C(9)>=1
THEN ERG$=ERG$&R$(43)&"+" EL
SE 185Ø
184Ø C(8),C(9)=Ø
185Ø IF C(9)>=1 AND C(13)>=1
 THEN ERG$=ERG$&R$(44)&"+" E
LSE 187Ø
186Ø C(9),C(13)=Ø
187Ø IF C(13)>=1 AND C(12)>=
1 THEN ERG$=ERG$&R$(45)&"+"
ELSE 189Ø
188Ø C(13),C(12)=Ø
189Ø IF C(Ø)>=1 AND C(4)>=1
THEN ERG$=ERG$&R$(46)&"+" EL
SE 191Ø
19ØØ C(Ø),C(4)=Ø
191Ø IF C(2)>=1 AND C(6)>=1
THEN ERG$=ERG$&R$(47)&"+" EL
SE 193Ø
192Ø C(2),C(6)=Ø
193Ø IF C(1Ø)>=1 AND C(14)>=
1 THEN ERG$=ERG$&R$(48)&"+"
ELSE 195Ø
194Ø C(1Ø),C(14)=Ø
195Ø IF C(8)>=1 AND C(12)>=1
 THEN ERG$=ERG$&R$(49)&"+" E
LSE 197Ø
196Ø C(8),C(12)=Ø
197Ø IF C(Ø)>=1 AND C(2)>=1
THEN ERG$=ERG$&R$(5Ø)&"+" EL
SE 199Ø
198Ø C(Ø),C(2)=Ø
199Ø IF C(1)>=1 AND C(3)>=1
THEN ERG$=ERG$&R$(51)&"+" EL
SE 2Ø1Ø
2ØØØ C(1),C(3)=Ø
2Ø1Ø IF C(5)>=1 AND C(7)>=1
THEN ERG$=ERG$&R$(52)&"+" EL
SE 2Ø3Ø
2Ø2Ø C(5),C(7)=Ø
2Ø3Ø IF C(4)>=1 AND C(6)>=1
THEN ERG$=ERG$&R$(53)&"+" EL
SE 2Ø5Ø
2Ø4Ø C(4),C(6)=Ø
2Ø5Ø IF C(2)>=1 AND C(1Ø)>=1
 THEN ERG$=ERG$&R$(54)&"+" E
LSE 2Ø7Ø
2Ø6Ø C(2),C(1Ø)=Ø
2Ø7Ø IF C(3)>=1 AND C(11)>=1
 THEN ERG$=ERG$&R$(55)&"+" E
LSE 2Ø9Ø
2Ø8Ø C(3),C(11)=Ø
2Ø9Ø IF C(7)>=1 AND C(15)>=1
 THEN ERG$=ERG$&R$(56)&"+" E
LSE 211Ø
21ØØ C(7),C(15)=Ø
211Ø IF C(6)>=1 AND C(14)>=1
 THEN ERG$=ERG$&R$(57)&"+" E
LSE 213Ø
212Ø C(6),C(14)=Ø
213Ø IF C(1Ø)>=1 AND C(8)>=1
 THEN ERG$=ERG$&R$(58)&"+" E
LSE 215Ø
214Ø C(1Ø),C(8)=Ø
215Ø IF C(11)>=1 AND C(9)>=1
 THEN ERG$=ERG$&R$(59)&"+" E
LSE 217Ø
216Ø C(11),C(9)=Ø
217Ø IF C(15)>=1 AND C(13)>=
1 THEN ERG$=ERG$&R$(6Ø)&"+"
ELSE 219Ø
```

```
2180 C(15),C(13)=0
2190 IF C(14)>=1 AND C(12)>=
1 THEN ERG$=ERG$&R$(61)&"+"
ELSE 2210
2200 C(14),C(12)=0
2210 IF C(0)>=1 AND C(8)>=1
THEN ERG$=ERG$&R$(62)&"+" EL
SE 2230
2220 C(0),C(8)=0
2230 IF C(1)>=1 AND C(9)>=1
THEN ERG$=ERG$&R$(63)&"+" EL
SE 2250
2240 C(1),C(9)=0
2250 IF C(5)>=1 AND C(13)>=1
 THEN ERG$=ERG$&R$(64)&"+" E
LSE 2270
2260 C(5),C(13)=0
2270 IF C(4)>=1 AND C(12)>=1
 THEN ERG$=ERG$&R$(65)&"+" E
LSE 2300
2280 C(4),C(12)=0
2290 !
2300 !1er BLOCK
2310 FOR I=0 TO 15 :: IF C(I
)=1 THEN ERG$=ERG$&R$(66+I)&
"+"
2320 NEXT I
2330 ERG$=SEG$(ERG$,1,LEN(ER
G$)-1)
2340 !
2350 !ERGEBNISAUSGABE
2360 A$="ERGEBNIS:" :: FOR I
=1 TO LEN(A$):: CALL HCHAR(2
0,16+I,ASC(SEG$(A$,I,1)))::
NEXT I
2370 ERG$="A="&ERG$ :: IF LE
N(ERG$)>28 THEN 2390
2380 FOR I=1 TO LEN(ERG$)::
CALL HCHAR(22,3+I,ASC(SEG$(E
RG$,I,1))):: NEXT I :: GOTO
2420
2390 TEIL2$=SEG$(ERG$,POS(ER
G$,"+",25)+1,(LEN(ERG$)-POS(
ERG$,"+",25))):: TEIL1$=SEG$
(ERG$,1,POS(ERG$,"+",25))
2400 FOR I=1 TO LEN(TEIL1$):
: CALL HCHAR(22,3+I,ASC(SEG$
(ERG$,I,1))):: NEXT I
2410 FOR I=1 TO LEN(TEIL2$):
: CALL HCHAR(24,3+I,ASC(SEG$
(TEIL2$,I,1))):: NEXT I
2420 A$="NEUE EINGABE ? J/N"
 :: FOR I=1 TO LEN(A$):: CAL
L HCHAR(17,14+I,ASC(SEG$(A$,
I,1))):: NEXT I
2430 CALL KEY(0,K,S):: IF S=
0 THEN 2430
2440 IF K=78 THEN 2450 ELSE
IF K=74 THEN 260 ELSE 2430
2450 CALL CLEAR :: END
```

```
100 !************************
110 !*     HEXDUMP            *
120 !* Copyright 1985 by      *
130 !*  Bernard Schmidt       *
140 !*                        *
150 !* Benoetigte Geraete :   *
160 !*  TI99/4A Konsole       *
170 !* TI Extended Basic      *
180 !*                        *
190 !* Speicherbelegung :     *
200 !*    1633 Bytes          *
210 !*                        *
220 !************************
230 CALL CLEAR
240 DISPLAY AT(8,3)BEEP:"HEX
DUMP DRUCKERAUSGABE "
250 DISPLAY AT(12,3):"FILENA
ME......:" :: ACCEPT AT(12,1
9):DRUCK$
260 IF DRUCK$="" THEN 250
270 DISPLAY AT(14,3):"ANFANG
ADRESSE :>" :: ACCEPT AT(14,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):ANF$
280 DISPLAY AT(16,3):"ENDADR
ESSE....:>" :: ACCEPT AT(16,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):END$
290 OPEN #1:DRUCK$ :: PRINT
#1:"......HEXDUMP" :: PRINT
#1 :: PRINT #1
300 PRINT #1:"......VON..>";
ANF$;"..BIS..>";END$
310 PRINT #1 :: PRINT #1 ::
CLOSE #1
320 CALL HEXDEZ(ANF,ANF$)::
CALL HEXDEZ(ENDE,END$)
330 DIF=ENDE-ANF :: X=INT(DI
F/16):: K=0 :: ADR=ANF
340 FOR I=1 TO X :: A$="" ::
 FOR J=0 TO 15 :: CALL PEEK(
ANF+K+J,WERT)
350 CALL DEZHEX(WERT,WERT$):
: A$=A$&" "&SEG$(WERT$,3,2):
: NEXT J
360 CALL DEZHEX(ADR,ADR$)
370 OPEN #1:DRUCK$ :: PRINT
#1:" >";ADR$;"..";A$ :: CLOS
E #1
380 K=K+16 :: ADR=ADR+16 ::
NEXT I
390 IF DIF-(X*16)>=0 THEN 40
0 ELSE 450
400 Y=DIF-(X*16):: A$="" ::
FOR I=0 TO Y
410 CALL PEEK(ANF+K+I,WERT):
: CALL DEZHEX(WERT,WERT$)
420 A$=A$&" "&SEG$(WERT$,3,2
):: NEXT I
430 CALL DEZHEX(ANF+K,ADR$)
440 OPEN #1:DRUCK$ :: PRINT
#1:" >";ADR$;"..";A$ :: CLOS
E #1
450 CALL CLEAR
460 DISPLAY AT(8,3)BEEP:"NEU
E EINGABE (J/N) ?"
470 CALL KEY(0,K,S):: IF S=0
 THEN 470
480 IF K=74 THEN 230 ELSE IF
 K=78 THEN END ELSE 470
490 SUB HEXDEZ(DEZ,HEX$)
500 K(1)=4096 :: K(2)=256 ::
 K(3)=16 :: K(4)=1
510 ZAHL$="0123456789ABCDEF"
520 DEZ=0 :: HEX$="0000"&HEX
$ :: HEX$=SEG$(HEX$,LEN(HEX$
)-3,4)
530 FOR N=1 TO 4 :: DEZ=(POS
(ZAHL$,SEG$(HEX$,N,1),1)-1)*
K(N)+DEZ
540 NEXT N :: SUBEND
550 SUB DEZHEX(DEZ,HEX$)
560 ZAHL$="0123456789ABCDEF"
 :: HEX$,HEX1$="" :: DEZ1=DE
Z
570 FOR N=1 TO 4 :: R=(DEZ1-
INT(DEZ1/16)*16):: DEZ1=INT(
DEZ1/16)
580 HEX1$=HEX1$&SEG$(ZAHL$,R
+1,1):: NEXT N
590 FOR N=1 TO 4 :: HEX$=HEX
$&SEG$(HEX1$,5-N,1):: NEXT N
 :: SUBEND
```

```
HEXDUMP

VON  >0000  BIS  >0036


>0000   83 E0 00 24 83 E0 09 00 83 C0 0A 92 30 AA 04 60
>0010   07 E2 00 06 1E 00 04 60 00 7A 1E 00 04 60 00 78
>0020   04 60 04 B2 02 0D 78 00 02 0E 01 00 02 0F 8C 02
>0030   02 00 00 20 10 13 10
```

```
100 !************************
110 !* POKELISTENGENERATOR *
120 !* Copyright 1985  by  *
130 !*   Bernard Schmidt   *
140 !*                     *
150 !* benoetigte Geraete: *
160 !*  TI 99/4A Konsole   *
170 !*  TI Extended Basic  *
180 !*   32K-Erweiterung   *
190 !*                     *
200 !*  Speicherbelegung:  *
210 !*     1390 Bytes      *
220 !*                     *
230 !***********************
240 CALL CLEAR
250 DISPLAY AT(8,3)BEEP:"ERS
TELLUNG VON POKELISTEN" :: D
ISPLAY AT(10,8):"MIT DRUCKER
 !"
260 DISPLAY AT(16,3):"FILENA
ME     :" :: ACCEPT AT(16,19)
:DRUCK$ :: IF DRUCK$="" THEN
 260
270 DISPLAY AT(17,3):"PROGRA
MMNAME ?" :: ACCEPT AT(17,19
):NAME$
280 DISPLAY AT(18,3):"STARTA
DRESSE:  >" :: ACCEPT AT(18,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):ANF$
290 DISPLAY AT(20,3):"ENDADR
ESSE  :  >" :: ACCEPT AT(20,
19)SIZE(4)VALIDATE(DIGIT,"AB
CDEF"):END$
300 OPEN #1:DRUCK$ :: PRINT
#1:"PROGRAMMNAME :";NAME$ ::
 PRINT #1 :: PRINT #1:"POKEL
ISTE VON >";ANF$;" BIS >";EN
D$
310 PRINT #1
320 PRINT #1 :: CLOSE #1
330 CALL HEXDEZ(ANF,ANF$)::
CALL HEXDEZ(ENDE,END$)
340 DIF=ENDE-ANF :: X=INT(DI
F/20):: K=0
350 FOR I=1 TO X :: A$=""
360 FOR J=1 TO 20 :: CALL PE
EK(ANF+K+J-1,WERT):: A$=A$&"
 "&STR$(WERT):: NEXT J
370 OPEN #1:DRUCK$
380 PRINT #1:" DATA  ";A$
390 CLOSE #1
400 K=K+20 :: NEXT I
410 IF DIF-(X*20)>=0 THEN 42
0 ELSE 470
420 Y=DIF-(X*20):: A$="" ::
FOR I=0 TO Y
430 CALL PEEK(ANF+K+I,WERT):
: A$=A$&" "&STR$(WERT):: NEX
T I
440 OPEN #1:DRUCK$
450 PRINT #1:" DATA  ";A$
460 CLOSE #1
470 CALL CLEAR
480 DISPLAY AT(8,3)BEEP:"NEU
E EINGABE (J/N) ?"
490 CALL KEY(0,K,S):: IF S=0
 THEN 490
500 IF K=74 THEN 240 ELSE IF
 K=78 THEN END ELSE 490
510 SUB HEXDEZ(DEZ,HEX$)
520 K(1)=4096 :: K(2)=256 ::
 K(3)=16 :: K(4)=1
530 ZAHL$="0123456789ABCDEF"
540 DEZ=0 :: HEX$="0000"&HEX
$ :: HEX$=SEG$(HEX$,LEN(HEX$
)-3,4)
550 FOR N=1 TO 4 :: DEZ=(POS
(ZAHL$,SEG$(HEX$,N,1),1)-1)*
K(N)+DEZ
560 NEXT N :: SUBEND
```

```
ROGRAMMNAME :PROBE

POKELISTE VON >0000 BIS >0024

 DATA   131 224 0 36 131 192 9 0 131 192 10 146 48 170 4 96 2 178 [illegible]
 DATA   30 0 4 96 0 122 30 0 4 96 0 120 4 96 4 178 2
```

SPASS SPIEL SPANNUNG

IRVEL

Mit Ihrem Raumschiff haben Sie sich auf dem Planeten Irvel verirrt. Um nach dem richtigen Heimweg zu fragen, verlassen Sie Ihr Schiff und erkunden den Planeten zu Fuß. Dabei geraten Sie in das gefürchtete Labyrinth von Irvel.

Zwar finden Sie hier reichlich Schätze vor, mit denen Sie Punkte sammeln können, doch nach 2000 Punkten wird es den Hütern der Schätze zu viel und sie versuchen, den Eindringling zu vernichten.

Die Mauern des Labyrinthes verschieben sich plötzlich und gefährliche Fallgruben entstehen.

Trotzdem kann das Labyrinth erst verlassen werden, wenn mindestens Schätze im Wert von 10 000 Punkten gesammelt wurden. Je höher die Punktzahl steigt, desto gefährlicher wird der Aufenthalt im Labyrinth.

Um Ihnen den Aufenthalt etwas zu erleichtern, bekommen Sie bei 4 500 Punkten einen Bonus und ein zusätzliches Leben.

Nur wenige sind aus diesem Labyrinth wieder heil ans Tageslicht gekommen.

Ob Sie nun auch dazu gehören werden ???

P.S.: Denkt daran: Die Hilfspunkte im Listing nicht mit abtippen.

H.K.

The Castle of 81 Rooms

Dieses Spiel ist im Adventure-Stil aufgebaut und hat eine 3D-Graphik. Die Aufgabe ist ganz einfach: Sie müssen aus dem Labyrinth herausfinden. Das müssen Sie aber lebendig bewerkstelligen. In der Burg mit den 81 Räumen (9x9) gibt es auch böse Geister und einen sogenannten ‚Burgvirus'. Der Kontakt mit einem von beiden endet für Sie tödlich, es sei denn, Sie haben vorher in einem der Räume eine Bibel gegen den Geist oder eine Pille aufgenommen. Nehmen Sie einen Gegenstand aus einem Raum auf, so verschwindet er von dort. Sie können jeweils nur drei Pillen und drei Bibeln tragen. Bei jedem ‚Feindkontakt' wird Ihnen das jeweilige Gegenmittel abgezogen. Truhen und Kelche voller Energie können Sie nehmen, soviel Sie wollen. Sinkt die Energie unter Null oder begegnen Sie einem Geist oder einem Virus ohne das entsprechende Gegenmittel, so sterben Sie.

G.S.

Ecarte

Dieses Programm erlaubt es, gegen den Computer Ecarte zu spielen. Hierbei gelten folgende Spielregeln: Es wird mit einem Skatspiel (32 Karten) gespielt. Die Reihenfolge ist: König, Dame, Bube, As, 10, 9, 8, 7.

Zuerst wird per Zufall bestimmt, welcher der beiden Spieler Geber und welcher Ausspielender ist. Dann werden die Karten gegeben (dies wird unabhängig von den Spielerbezeichnungen immer vom Computer erledigt.). Jeder erhält fünf Karten. Die elfte Karte wird offen auf den Tisch gelegt und bestimmt die Trumpffarbe. Ist es ein König, so erhält der Geber einen Punkt. Auch wenn einer der beiden Spieler den Trumpfkönig hat oder ihn später erhält, so kann er ihn anmelden und erhält einen Punkt dafür. (Taste ‚a').

Nun muß der Ausspielende entscheiden, ob er spielen will (Taste ‚s') oder ob er einige Karten austauschen will (Taste ‚k'). Will er Karten austauscben, so muß der Geber entscheiden, ob er doch spielen soll oder die Karten tauschen kann. dies wird wieder durch dieselben Tasten gesteuert. Bestätigt er den Austausch, so müssen zuerst der Ausspielende und dann der Geber die Zahl und die Nummern der auszutauschenden Karten angeben. Hierbei denkt man sich die Karten von oben nach unten von 1-5 durchnumeriert. Dies ist nur möglich, solange noch Karten auf dem Talon liegen.

Hat einer der Spieler entschieden, daß das Spiel beginnen soll, so spielt der Ausspielende die erste Karte auf. Von diesem Zeitpunkt an ist es nicht mehr möglich, enen Trumpfkönig anzumelden. Für das Bedienen gelten folgende Regeln:

– Man muß, sofern möglich, immer bedienen!
– Jeder Stich muß, wenn möglich, mitgenommen werden!

Eine Karte wird durch Eingabe ihrer Nummer ausgespielt. Nach fünf Stichen ist das Spiel zuende.

Hat derjenige, der zum Spiel aufgefordert hat, alle Stiche gemacht, so bekommt er zwei Punkte; hat er mindestens drei Stiche gemacht, so erhält er einen Punkt und hat er weniger als drei, werden ihm zwei Punkte abgezogen.

Der Computer überwacht, daß der Spieler die Regeln genau einhält und spielt den Stich selbst, wenn man das nicht tut.

Sky-Driver

Sky-Diver ist ein Spiel, das auf dem TI-Grundgerät läuft. Aufgabe des Spielers ist es, Fallschirmspringer aus einem Hubschrauber so abzusetzen, daß diese die Landemarke möglichst in der Mitte treffen.

Die Position der Landemarke sowie der Hindernisse (Bäume und eine große Wolke) werden

durch die RND-Funktion bestimmt.

Anfangs stehen 5 Springer zur Verfügung. Wird die Landemarke in der Mitte getroffen, so erhält der Spieler 3 Punkte; die Anzahl der Springer bleibt unverändert. Wird die Marke am Rand getroffen, so erhält man einen Punkt, aber es wird ein Springer abgezogen. Wird die Marke nicht getroffen, so wird ebenfalls ein Springer abgezogen.

Bei 30 erreichten Punkten bekommt man einen Bonus-Springer. Der Spieler kann zwischen einer leichteren Spielvariante (ohne Windbeeinflussung des Springers) und einer etwas schwierigeren wählen.

Variablen-Liste

I,K,L	Schleifenvariable
M,Y	Merker für Auszuführende bzw. nicht mehr auszuführende Programmteile
A,B	zum Einlesen der Data-Zeilen
SP	Spielart
P	Punkte
MEN	Anzahl der noch vorhandenen Springer
BZ	Merker, ob der Bonus-Springer bereits einmal zugezählt wurde
WM	Wind von links/ rechts bzw. kein Wind
Z,Q,W	Variablen für die GChar-Abfrage
L,U	Bestimmen die Sprungbewegung
PU	Abfrage ob 0,1 oder 3 Punkte erzielt wurden
T	Bestimmt, ob ein Spieler abgezogen wird oder nicht
EOF	

Dirk Junghans

CATCH WITH CAR

Catch with Car ist ein Spielprogramm für den Ti 99/4A in der Grundausstattung. Um das Programm laufen zu lassen, werden folgende Hardware-„Teile" benötigt:
I Ti99/4A Computer
1 Joystik (Joystik Nr. 2)
I Fernseher/Monitor
und weiter empfehle ich einen Cassettenrekorder oder eine Diskettenstation zum Abspeichern des Programms.

Sie leben auf einem fremden Planeten, der von riesigen Insekten beherrscht wird. Ihre Aufgabe ist es nun, mit Joystik 2 ein Fahrzeug zu steuern und dabei die Insekten, die immer vereinzelt auftauchen, zu überfahren. Da es auf dem Planeten ausschließlich Sandbogen gibt, reißt Ihr Fahrzeug tiefe Furchen in den Sand. Wenn Sie nun in eine solche Furche hineinfahren, gibt es Punktverlust. Insgesamt dürfen Sie auch nur 15-mal in eine solche Furche fahren.

Es gibt verschiedene Arten von Insekten auf dem Planeten, bei einigen gibt es nur Punkte, wenn sie diese überfahren, bei anderen verschwindet an einigen vereinzelten Stellen die Furche und bei noch anderen gibt es Punkte und es verschwinden Furchenabschnitte. Ihnen stehen drei Fahrzeuge zur Verfügung, diese Zahl vermindert sich, wenn Sie aus dem Spielfeld fahren. Folgende Punkte führen zum Spielende:

Mehr als 15mal in eine Furche gefahren, weniger als Null Punkte. Dreimal aus dem Spielfeld gefahren.

Wenn Sie 1500 Punkte erreicht haben, erhöht sich die Anzahl der Fahrzeuge um eins, 25 Furchenabschnitte werden gelöscht und es ertönt eine Melodie.

Zu Beginn des Spiels und wenn Sie aus dem Feld gefahren sind, müssen Sie den Feuerknopf betätigen, um weiterfahren zu können. Und nun wünsche ich Ihnen viel Spaß mit dem Spiel!!!

Dirk Junghans

Dreiecks- und Kreisberechnungen

Mit diesem Programm stellt Euch der Autor, Jürgen Brühne, eine kleine Hilfe für die Hausaufgaben zur Verfügung. Seid Ihr in Mathe gerade bei dem Thema Geometrie angelangt, kann der Rechner nun Eure Hausaufgaben kontrollieren. Natürlich sollten die Aufgaben auch weiterhin alleine gemacht werden, denn wenn der TI99 Eure Aufgaben alleine macht, bleibt von dem Stoff bei Euch wenig hängen und am Ende des Schuljahres schlägt sich dies in Eurem Notenspiegel nieder. Auf jeden Fall dürft Ihr Euren Rechner aber die Aufgaben auf korrekte Lösungen überprüfen lassen, um am nächsten Tag nicht auf die Nase zu fallen. Da das Programm vollständig in TI-Basic geschrieben wurde, braucht keiner von Euch darauf zu verzichten.

Ein Menue führt durch das Programm und es müssen nur noch die einzelnen Werte der Berechnungen eingegeben werden.

Jürgen Brühne

Dreidimensionale Mühle

Der Sinn dieses Spieles ist es, möglichst viele „Mühlen" zu bauen. 2 Spieler erhalten je 32 Steine, die sie abwechselnd setzen — einer dieser Spieler kann auch der Computer sein — Man versucht nun, durch geschicktes Setzen der Steine jeweils 4 in eine senkrechte, waagerechte oder diagonale Linie zu bringen:

```
0              0
0            0
0  0000    0
0        0
```

Dies sind die Mühlen. Wenn alle Steine gesetzt sind, hat der Spieler mit den meisten Mühlen gewonnen. Das Zählen der Mühlen übernimmt der Computer.

Bei diesem Spiel müssen einmal nicht feindliche Objekte abgeschossen werden; es soll vielmehr das logische und räumliche Denken gefördert werden. *Jürgen Brühne*

Copyright Zeichen

Das Programm „Copyright Zeichen" ist in TI-Basic programmiert worden, läuft aber auch unter Extended Basic!

Sicher haben Sie auch schon einmal überlegt, welches Copyrightzeichen Sie für Ihr Programm verwenden wollen, und haben dann krampfhaft irgendein Zeichen definiert. Mit diesem Programm hat das ein Ende! Hier haben Sie die Auswahl zwischen 11 vordefinierten Copyrightzeichen. Wählen Sie in einem Menü Ihr Zeichen und der TI wird Ihnen den 16stelligen Code auf den Bildschirm schreiben! Viel Spaß beim Aussuchen. *Matthias Orf*

```
10 REM********************
11 REM*                  *
12 REM*     IRVEL        *
13 REM*  Copyright  by   *
14 REM* Manfred Lipowski *
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*   12779 Bytes    *
25 REM********************
26 REM
120 RANDOMIZE
130 CALL CLEAR
140 CALL SCREEN(7)
150 CALL COLOR(1,16,16)
160 FOR A=1 TO 24
170 PRINT
180 NEXT A
190 REM
        GRAFIK IRVEL
200 CALL CHAR(64,"3C4299A1A1
99423C")
210 CALL CHAR(96,"0018180018
181818")
220 CALL CHAR(97,"1818181818
181818")
230 CALL CHAR(98,"007C7E6666
66667E")
240 CALL CHAR(99,"7C786C6666
666666")
250 CALL CHAR(100,"006666666
6666666")
260 CALL CHAR(101,"663C3C3C3
C181818")
270 CALL CHAR(102,"007E7E606
060607C")
280 CALL CHAR(103,"7C6060606
0607E7E")
290 CALL CHAR(104,"006060606
0606060")
300 CALL CHAR(105,"606060606
0607E7E")
310 REM
        GRAFIK SETZEN
320 CALL COLOR(9,5,16)
330 CALL COLOR(10,5,16)
340 FOR A=2 TO 8
350 CALL COLOR(A,13,16)
360 NEXT A
370 PRINT ".......@ ' b d f
h.......":".........a c e g
i"
380 PRINT : : : : : : : : : :
:
390 PRINT "..PRESS ANY KEY T
O BEGIN"
400 DATA 262,330,392,262,330
,392,349,440,523,349,440,523
,262,330,392,262,330,392
410 DATA 196,247,294,196,247
,294,262,330,392,262,330,392
,349,440,523,349,440,523
420 DATA 262,330,392,196,247
,294,262,262,262
430 RESTORE 400
440 FOR A=1 TO 45
450 READ BOT
460 CALL KEY(0,K,S)
470 IF S=0 THEN 480 ELSE 510
480 CALL SOUND(-999,BOT,0,BO
T+2,0,BOT-2,0)
490 NEXT A
500 GOTO 430
510 GOSUB 3250
520 REM   SPIELERKLAERUNG
              ODER
          PROGRAMANLEITUNG
530 FOR A=1 TO 24
540 PRINT
550 NEXT A
560 PRINT "..SPIELERKLAERUNG
 ? (J/N)"
570 PRINT : : : : : : : : : :
:
580 GOTO 600
590 CALL SOUND(99,110,0,220,
0)
600 CALL KEY(0,K,S)
610 IF S=0 THEN 600
620 IF K=78 THEN 820
630 IF K=74 THEN 650
640 GOTO 590
650 PRINT ".......@ ' b d f
h":".........a c e g i": :
660 PRINT "IHRE AUFGABE BEST
EHT DARIN": :"DEN ABENTEURER
 SICHER DURCH": :"DAS LABYRI
NTH VON IRVEL ZU": :"FUEHREN
."
670 PRINT :"ER MUSS ALLE 10
SCHAETZE": :"BERUEHREN UM IN
 DIE NAECHSTE": :"RUNDE ZU K
OMMEN. .......>>": : : : :
680 CALL KEY(0,K,S)
690 IF S=0 THEN 680
700 PRINT :"BEI EINEM PUNKTE
STAND VON": :"2000 P WERDEN
DIE TEMPEL-": :"GOETTER ZORN
IG UND FANGEN": :
710 PRINT "AN DAS LABYRINT
ZU VER-": :"SCHIEBEN.MAUERN
WERDEN EIN-": :"GERISSEN UND
 AN ANDERER": :"STELLE WIEDE
R AUFGEBAUT !"
720 PRINT :"LOECHER ERSCHEIN
EN IM BODEN": :"UND VERSCHWI
NDEN NACH EINER": :"WEILE.ER
ST WENN IHR P-KONTO": :
730 PRINT "10000 P ANZEIGT W
```

```
ERDEN SIE": :"AUS DEM LABYRI
NTH ENTLASSEN."
74Ø CALL KEY(Ø,K,S)
75Ø IF S=Ø THEN 74Ø
76Ø PRINT :"DAS SPIEL WIRD M
IT JOYSTICK": :"1 GESTEUERT.
DER ABENTEURER": :"KANN IN J
EDE RICHTUNG ZIEHEN": :
77Ø PRINT ",DIAGONAL,SENKREC
HT U.S.W.": :"WIRD IHR ABENT
EURER WOHL JE": :"WIEDER DEN
 BLAUEN HIMMEL VON": :"IRVEL
 ZU SEHEN BEKOMMEN ?"
78Ø PRINT :"WER WEISS...!":
: :"ALPHA LOCK BITTE AUSRAST
EN !"
79Ø PRINT : : : :
8ØØ CALL KEY(Ø,K,S)
81Ø IF S=Ø THEN 8ØØ
82Ø CALL CLEAR
83Ø CALL SCREEN(2)
84Ø REM
            GRAFIK FUER EINGEN
DLICHES     PROGRAMM
            WAENDE
85Ø CALL CHAR(33,"3C7EFFFFFF
FF7E3C")
86Ø CALL CHAR(4Ø,"FF888888FF
2424FF")
87Ø CALL CHAR(41,"FF212121FF
8989FF")
88Ø CALL CHAR(42,"FF848484FF
9191FF")
89Ø REM
            GRAFIK SCHAETZE
9ØØ CALL CHAR(112,"1F2345FDB
58586FC")
91Ø CALL CHAR(113,"224149556
3777F3E")
92Ø CALL CHAR(114,"1Ø1Ø381Ø1
Ø1Ø1Ø1Ø")
93Ø CALL CHAR(115,"7F3ØFC783
ØØØØØØØ")
94Ø CALL CHAR(116,"ØØØØFEØ4Ø
E1FØEØ4")
95Ø CALL CHAR(117,"183C5A7E7
E243C18")
96Ø CALL CHAR(118,"1C222A5D4
92A221C")
97Ø CALL CHAR(119,"ØØØØ4ØAØB
FA545ØØ")
98Ø REM   ABENTEURER
99Ø CALL CHAR(12Ø,"181866BD9
9242442")
1ØØØ CALL CHAR(121,"181A6FBD
9A242412")
1Ø1Ø CALL CHAR(122,"1858F6BD
59242448")
1Ø2Ø REM GRAFIK ABENTEU.ANZE
IGE U. HINTERG.
1Ø3Ø CALL CHAR(14Ø,"ØØØØØØØØ
ØØØØØØØØ")
1Ø4Ø CALL CHAR(141,"181866BD
99242442")
1Ø5Ø REM   SPIELFELD AUFBAU
1Ø6Ø CALL COLOR(2,13,15)
1Ø7Ø CALL COLOR(1,2,16)
1Ø8Ø CALL COLOR(14,15,1)
1Ø9Ø CALL COLOR(3,8,1)
11ØØ CALL COLOR(4,8,1)
111Ø CALL COLOR(5,11,1)
112Ø CALL COLOR(6,11,1)
113Ø CALL COLOR(7,11,1)
114Ø CALL COLOR(8,11,1)
115Ø CALL COLOR(11,5,16)
116Ø CALL COLOR(12,2,16)
117Ø CALL HCHAR(1,1,14Ø,96)
118Ø DATA "(((((((((((((((((
(((((((((((((((","() *) (...
.(....*)....(...( *) ("
119Ø DATA "()....(....(....*
)....(...(....(","(((() ( *(
(((() *) *(((() ( *(((","().
.............*).............
("
12ØØ DATA "(((((((() *((((((
((((((() *(((((","()........
......*).............("
121Ø DATA "((() *((((()..( *
) (..*(((() *((","()........
.( () *) *) (........(","(((
(((((() (..(....(..( *((((((
("
122Ø DATA "().........*(((((
((((()........(","() *((((((
(((()....*((((((((() ("
123Ø DATA "().........(..(..
..(..(........(","((() *((((
() *() *(() *(((() *(("
124Ø DATA "()...............
..............(","(((((((()
*(((((((((((() *(((((("
125Ø DATA "()..............*
).............(","(((() ( *(
(((() *) *(((() ( *((("
126Ø DATA "()....(...(.....*
)....(...(....(","() *) (...
(.....*)....(...( *) ("
127Ø DATA "(((((((((((((((((
(((((((((((((((("
128Ø RESTORE 118Ø
129Ø FOR A=4 TO 24
13ØØ READ B$
131Ø FOR B=1 TO LEN(B$)
132Ø CALL HCHAR(A,B,ASC(SEG$
(B$,B,1)))
133Ø NEXT B
134Ø NEXT A
135Ø A=1
136Ø B=3
137Ø A$="SCORE:"
138Ø GOSUB 289Ø
```

```
1390 A=1
1400 B=19
1410 A$="MEN:"
1420 GOSUB 2890
1430 MEN=3
1440 GOSUB 2940
1450 SCORE=50
1460 GOSUB 3040
1470 GOSUB 2390
1480 REM  ABENTEUERER ZUFAELL
IG        SETZEN !
1490 XX=INT(RND*19)+4
1500 YY=INT(RND*27)+2
1510 CALL GCHAR(XX,YY,PP)
1520 IF PP<>32 THEN 1490
1530 CALL HCHAR(XX,YY,120)
1540 GOSUB 3250
1550 PAT=120
1560 REM  HAUPTTEIL DES
          PROGRAMMS
1570 CALL HCHAR(XX,YY,PAT)
1580 CALL SOUND(-999,-4,9)
1590 IF SCORE<=2000 THEN 162
0 ELSE 1600
1600 IF RND<.4 THEN 1610 ELS
E 1620
1610 GOSUB 2520
1620 CALL JOYST(1,X,Y)
1630 IF (X=0)*(Y=0)THEN 1570
1640 IF X=-4 THEN 1690
1650 IF X=4 THEN 1720
1660 IF Y=-4 THEN 1750
1670 IF Y=4 THEN 1780
1680 GOTO 1620
1690 PAT=121
1700 SOU=-5
1710 GOTO 1800
1720 PAT=122
1730 SOU=-5
1740 GOTO 1800
1750 PAT=120
1760 SOU=-6
1770 GOTO 1800
1780 PAT=120
1790 SOU=-7
1800 CALL HCHAR(XX,YY,32)
1810 SCORE=SCORE-1
1820 IF SCORE<0 THEN 3310
1830 XX=XX+(Y/-4)
1840 YY=YY+(X/4)
1850 CALL GCHAR(XX,YY,PP)
1860 IF PP<>32 THEN 1900
1870 CALL HCHAR(XX,YY,PAT)
1880 CALL SOUND(-1,SOU,9)
1890 GOTO 1620
1900 REM    ABFRAGE OB WAND
 ODER SCHATZ BERUEHRT
1910 IF (PP=40)+(PP=41)+(PP=
42)+(PP=33)THEN 1920 ELSE 21
10
1920 FOR A=3 TO 30 STEP 3
1930 CALL SOUND(-99,-5,A)
1940 NEXT A
1950 CALL HCHAR(XX,YY,40)
1960 MEN=MEN-1
1970 A=3
1980 B=3
1990 A$="DORT"&CHR$(140)&"WA
R"&CHR$(140)&"EINE"&CHR$(140
)&"MAUER"
2000 IF PP=33 THEN 2010 ELSE
 2020
2010 A$="INS"&CHR$(140)&"LOC
H"&CHR$(140)&"GEFALLEN"
2020 GOSUB 2890
2030 GOSUB 3250
2040 GOSUB 2940
2050 SCORE=SCORE-100
2060 GOSUB 3040
2070 FOR A=1 TO 400
2080 NEXT A
2090 CALL HCHAR(3,1,140,32)
2100 GOTO 1490
2110 CALL HCHAR(XX,YY,PAT)
2120 TONO=INT(RND*220)+220
2130 FOR A=0 TO 30 STEP 5
2140 CALL COLOR(12,7,16)
2150 CALL SOUND(-300,TONO,A,
TONO+2,A,TONO/2,A)
2160 CALL COLOR(12,2,16)
2170 NEXT A
2180 IF PP=117 THEN 2190 ELS
E 2210
2190 SCORE=SCORE+150
2200 GOTO 2320
2210 IF PP=113 THEN 2220 ELS
E 2240
2220 SCORE=SCORE+125
2230 GOTO 2320
2240 IF PP=118 THEN 2250 ELS
E 2270
2250 SCORE=SCORE+200
2260 GOTO 2320
2270 IF PP=112 THEN 2280 ELS
E 2310
2280 SCOR1=INT(RND*150)+25
2290 SCORE=SCORE+SCOR1
2300 GOTO 2320
2310 SCORE=SCORE+100
2320 GOSUB 3040
2330 SCOR1=0
2340 SCHATZ=SCHATZ+1
2350 IF SCHATZ=10 THEN 2360
ELSE 1620
2360 GOSUB 2390
2370 GOTO 1570
2380 REM   SCHATZ SETZEN
2390 FOR AP=1 TO 10
2400 QWE=INT(RND*220)+220
2410 QQ=INT(RND*19)+4
```

```
2420 WW=INT(RND*29)+2
2430 CALL GCHAR(QQ,WW,EE)
2440 IF EE<>32 THEN 2410
2450 RR=INT(RND*8+1)+111
2460 CALL HCHAR(QQ,WW,RR)
2470 CALL SOUND(-99,QWE,AP*2
,QWE/2,AP*2)
2480 NEXT AP
2490 SCHATZ=0
2500 RETURN
2510 REM     WAND ENTFERNEN
             LOCH SETZEN
             LOCH ENTFERNEN
             WAND SETZEN
2520 REM
2530 ZX=INT(RND*18)+5
2540 XZ=INT(RND*27)+3
2550 CALL GCHAR(ZX,XZ,ZZ)
2560 IF (ZZ<>40)*(ZZ<>41)*(Z
Z<>42)THEN 2530
2570 CALL SOUND(-99,-7,0)
2580 CALL HCHAR(ZX,XZ,32)
2590 CALL SOUND(-99,-5,3)
2600 ON INT(RND*3+1)GOTO 261
0,2680,2790
2610 ZX=INT(RND*18)+5
2620 XZ=INT(RND*27)+5
2630 CALL GCHAR(ZX,XZ,ZZ)
2640 IF ZZ<>32 THEN 2610
2650 CALL SOUND(-99,-5,0)
2660 CALL HCHAR(ZX,XZ,33)
2670 CALL SOUND(-99,-6,3)
2680 IF RND<.4 THEN 2690 ELS
E 2790
2690 ZX=INT(RND*18)+5
2700 XZ=INT(RND*27)+5
2710 XTA=XTA+1
2720 IF XTA=5 THEN 2780
2730 CALL GCHAR(ZX,XZ,ZZ)
2740 IF ZZ<>33 THEN 2690
2750 CALL SOUND(-99,-6,0)
2760 CALL HCHAR(ZX,XZ,40)
2770 CALL SOUND(-99,-7,5)
2780 XTA=0
2790 IF RND<.6 THEN 2800 ELS
E 2870
2800 ZX=INT(RND*18)+5
2810 XZ=INT(RND*27)+5
2820 CALL GCHAR(ZX,XZ,ZZ)
2830 IF ZZ<>32 THEN 2800
2840 CALL SOUND(-99,-1,0)
2850 CALL HCHAR(ZX,XZ,40)
2860 CALL SOUND(-99,-3,0)
2870 RETURN
2880 REM WOERTER PRINTEN
2890 FOR C=1 TO LEN(A$)
2900 CALL HCHAR(A,B+C,ASC(SE
G$(A$,C,1)))
2910 CALL SOUND(1,-1,0)
2920 NEXT C
2930 RETURN
2940 IF MEN<=0 THEN 2990 ELS
E 2950
2950 CALL HCHAR(1,24,140,4)
2960 CALL HCHAR(1,24,141,MEN
)
2970 CALL SOUND(-1,-2,6)
2980 RETURN
2990 CALL HCHAR(1,24,48)
3000 GOSUB 3040
3010 GOSUB 3250
3020 GOTO 3310
3030 REM SCORE PRINTEN
3040 CALL HCHAR(1,10,140,6)
3050 FOR AFF=1 TO LEN(STR$(S
CORE))
3060 CALL HCHAR(1,9+AFF,ASC(
SEG$(STR$(SCORE),AFF,1)))
3070 CALL SOUND(-1,-3,5)
3080 NEXT AFF
3090 IF SCORE<=0 THEN 3310
3100 IF (SCORE>4500)*(SCORE<
5500)THEN 3110 ELSE 3220
3110 SCORE=SCORE+2500
3120 A=3
3130 B=3
3140 A$="BONUS"&CHR$(140)&"B
ONUS"&CHR$(140)&"BONUS"&CHR$
(140)&"BONUS"
3150 GOSUB 2890
3160 MEN=MEN+1
3170 GOSUB 2940
3180 GOSUB 3250
3190 GOSUB 3250
3200 GOSUB 3250
3210 CALL HCHAR(3,1,140,32)
3220 IF SCORE>=10000 THEN 37
10
3230 RETURN
3240 REM    ZUFALLSERZEUGUNG
                   TON
3250 TON=INT(RND*280)+220
3260 FOR ATT=0 TO 30 STEP 2.
5
3270 CALL SOUND(-99,TON,ATT,
TON+2,ATT,TON/2,ATT)
3280 NEXT ATT
3290 RETURN
3300 REM    SPIELENDE
3310 FOR A=2 TO 28
3320 CALL COLOR(1,2,7)
3330 CALL COLOR(11,2,7)
3340 CALL SOUND(-99,400,A-2,
400+A*2,A,400+A*2,A)
3350 CALL COLOR(1,2,16)
3360 CALL COLOR(11,5,16)
3370 CALL SOUND(-99,400,A-2,
400-A*2,A,400-A*2,A)
3380 NEXT A
3390 FOR A=0 TO 30 STEP 2.5
3400 CALL SCREEN(7)
```

```
3410 CALL SOUND(-99,-5,A)
3420 CALL SCREEN(16)
3430 NEXT A
3440 CALL SCREEN(2)
3450 FOR A=32 TO 1 STEP -1
3460 CALL VCHAR(4,A,140,21)
3470 NEXT A
3480 CALL HCHAR(3,1,140,32)
3490 SCORE=SCORE+150
3500 GOSUB 3040
3510 A=8
3520 B=4
3530 A$="LEIDER"&CHR$(140)&"
VERLOREN"
3540 GOSUB 2890
3550 A=10
3560 B=4
3570 A$="NOCHMAL"&CHR$(140)&
"?"&CHR$(140)&"J"&CHR$(140)&
"N"
3580 GOSUB 2890
3590 RESTORE 400
3600 FOR A=1 TO 45
3610 READ BOT
3620 CALL SOUND(-999,BOT,0,B
OT+2,0,BOT+4,0)
3630 CALL KEY(0,K,S)
3640 IF S=0 THEN 3650 ELSE 3
670
3650 NEXT A
3660 GOTO 3590
3670 IF (K=74)+(K=106)THEN 1
050
3680 GOSUB 3250
3690 CALL CLEAR
3700 END
3710 REM             SPIEL
                   GESCHAFFT
3720 FOR A=1 TO 3
3730 FOR B=560 TO 760 STEP 2
0
3740 CALL SOUND(-99,560,A,B,
A,B+2,A)
3750 NEXT B
3760 NEXT A
3770 CALL CLEAR
3780 CALL SCREEN(2)
3790 CALL CHAR(136,"7FDB6E1F
03000000")
3800 CALL CHAR(138,"C060C0E0
B0D8340E")
3810 CALL CHAR(139,"03000000
00000000")
3820 CALL CHAR(128,"0000003C
66FBFEBF")
3830 CALL CHAR(129,"FF6E3C00
00000000")
3840 CALL CHAR(130,"0000003C
6EFFFF7D")
3850 CALL CHAR(131,"F37E3C00
00000000")
3860 CALL CHAR(104,"00000303
0F0F3CFC")
3870 CALL CHAR(105,"FFFFC3C3
00000000")
3880 CALL CHAR(106,"0000C0C0
F0F03C3F")
3890 CALL COLOR(14,16,5)
3900 CALL COLOR(13,11,5)
3910 CALL COLOR(10,15,5)
3920 CALL HCHAR(1,1,140,224)
3930 CALL HCHAR(2,4,136)
3940 CALL HCHAR(2,5,138)
3950 CALL HCHAR(3,5,139)
3960 CALL HCHAR(3,20,136)
3970 CALL HCHAR(3,21,138)
3980 CALL HCHAR(4,21,139)
3990 CALL HCHAR(2,28,128)
4000 CALL HCHAR(2,29,130)
4010 CALL HCHAR(3,28,129)
4020 CALL HCHAR(3,29,131)
4030 FOR A=8 TO 24
4040 CALL SOUND(-99,-7,A)
4050 CALL HCHAR(A,1,40,32)
4060 CALL HCHAR(A,32-A,32,2)
4070 NEXT A
4080 FOR A=1 TO 200
4090 NEXT A
4100 ZZ=0
4110 FOR A=24 TO 7 STEP -1
4120 CALL HCHAR(A,9+ZZ,122)
4130 CALL SOUND(2,-7,4)
4140 CALL HCHAR(A,9+ZZ,32)
4150 CALL SOUND(2,-5,1)
4160 ZZ=ZZ+1
4170 NEXT A
4180 CALL COLOR(12,2,5)
4190 CALL HCHAR(7,26,120)
4200 CALL SOUND(2,-6,0)
4210 FOR A=1 TO 99
4220 NEXT A
4230 CALL HCHAR(7,7,104)
4240 CALL HCHAR(7,8,105)
4250 CALL HCHAR(7,9,106)
4260 FOR A=0 TO 30 STEP 5
4270 CALL SOUND(-99,-5,A)
4280 NEXT A
4290 FOR A=26 TO 10 STEP -1
4300 CALL SOUND(3,-7,3)
4310 CALL HCHAR(7,A,121)
4320 CALL SOUND(3,-6,8)
4330 CALL HCHAR(7,A,140)
4340 NEXT A
4350 CALL HCHAR(6,8,120)
4360 CALL SOUND(9,-5,0)
4370 FOR A=8 TO 24
4380 CALL SOUND(-99,-5,A)
4390 CALL HCHAR(A,1,40,32)
4400 CALL SOUND(-99,-6,A)
4410 NEXT A
4420 CALL COLOR(4,2,13)
```

```
4430 CALL COLOR(5,2,13)
4440 CALL COLOR(6,2,13)
4450 CALL COLOR(7,2,13)
4460 CALL COLOR(8,2,13)
4470 A=12
4480 B=4
4490 A$="G(E(S(C(H(A(F(F(T"
4500 GOSUB 2890
4510 A=14
4520 B=4
4530 A$="NOCH(EIN(SPIEL(?(J(
N"
4540 GOSUB 2890
4550 DATA 220,294,294,330,37
0,294,370,330,220,294,294,33
0,370,292,277,277
4560 DATA 220,294,294,330,37
0,392,370,330,294,277,220,24
7,277,294,294
4570 RESTORE 4550
4580 FOR A=1 TO 31
4590 CALL COLOR(10,16,5)
4600 READ B
4610 CALL KEY(0,K,S)
4620 IF S=0 THEN 4630 ELSE 4
670
4630 CALL COLOR(10,15,5)
4640 CALL SOUND(-550,B,0,B+2
,0,B+2,0)
4650 NEXT A
4660 GOTO 4570
4670 IF (K=74)+(K=106)THEN 1
00
4680 CALL CLEAR
4690 END
```

BÖRSE


```
10 REM*******************
11 REM*                 *
12 REM* COPYRIGHT-ZEICHEN*
13 REM*  Copyright by   *
14 REM*  Matthias Orf    *
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*    3618 Bytes    *
25 REM*******************
26 REM
270 CALL CLEAR
280 CALL CHAR(48,"0038444C54
644438")
290 CALL CHAR(79,"0038444444
444438")
300 CALL CHAR(97,"3C4299A1A1
99423C")
310 CALL CHAR(98,"3C4299A199
423C00")
320 CALL CHAR(99,"FF81B9A1A1
B981FF")
330 CALL CHAR(100,"FF8199A1A
19981FF")
340 CALL CHAR(101,"3C4299919
199423C")
350 CALL CHAR(102,"3C4218202
018423C")
360 CALL CHAR(103,"004299A1A
1994200")
370 CALL CHAR(104,"001C20202
0201C00")
380 CALL CHAR(105,"003C20202
0203C00")
390 CALL CHAR(106,"E781B9A1A
1B981E7")
400 CALL CHAR(107,"E78199A1A
1A199E7")
410 FOR I=1 TO 24
420 PRINT
430 NEXT I
440 FOR E=1 TO 14
450 CALL COLOR(E,16,11)
460 NEXT E
470 CALL SCREEN(16)
480 PRINT ".....C O P Y R I
G H T": :".......Z E I C H E
 N": : : : : : : : : : : : :
 : : : : : :
490 X=12
500 Y=9
510 T$="MARO-SOFT"
520 GOSUB 2090
530 X=14
540 Y=9
550 T$="BAUNATAL"
560 GOSUB 2090
570 CALL COLOR(9,5,11)
580 CALL COLOR(10,5,11)
```

```
590 FOR I=107 TO 97 STEP -1
600 CALL HCHAR(12,8,I)
610 FOR A=1 TO 130
620 NEXT A
630 NEXT I
640 FOR I=1 TO 2
650 FOR W=200 TO 2000 STEP 3
0
660 CALL SOUND(-50,W,0)
670 NEXT W
680 FOR W=2000 TO 200 STEP -
20
690 CALL SOUND(-50,W,0)
700 NEXT W
710 NEXT I
720 FOR W=0 TO 30 STEP 1
730 CALL SOUND(-200,200,W)
740 NEXT W
750 X=18
760 Y=3
770 T$="SIE HABEN DIE QUAL D
ER"
780 GOSUB 2090
790 X=20
800 Y=2
810 T$="WAHL BEI DEN COPY-ZE
ICHEN"
820 GOSUB 2090
830 FOR I=1 TO 600
840 NEXT I
850 FOR E=1 TO 7
860 FOR W=2 TO 16
870 CALL SCREEN(W)
880 NEXT W
890 NEXT E
900 CALL CLEAR
910 FOR E=1 TO 14
920 CALL SCREEN(11)
930 CALL COLOR(E,7,11)
940 NEXT E
950 CALL COLOR(9,2,11)
960 CALL COLOR(10,2,11)
970 PRINT "...a b c d e f g
h i j  k": : :"...1 2 3 4 5
6 7 8 9 10 11": : : :
980 PRINT "WAEHLEN SIE JETZT
 BITTE IHR": :"...COPYRIGHT-
ZEICHEN UND": :" IHR TI-99/4
A HOMECOMPUTER": :
990 PRINT "SCHREIBT IHNEN DE
N CODE ZUR": :"VERWENDUNG IN
 IHREM EIGENEN": :"........P
ROGRAMM !": : :
1000 INPUT "IHRE WAHL?..(1-1
1).. ":W
1010 IF W=1 THEN 1150
1020 IF W=2 THEN 1210
1030 IF W=3 THEN 1270
1040 IF W=4 THEN 1330
1050 IF W=5 THEN 1390
1060 IF W=6 THEN 1450
1070 IF W=7 THEN 1510
1080 IF W=8 THEN 1570
1090 IF W=9 THEN 1630
1100 IF W=10 THEN 1690
1110 IF W=11 THEN 1750
1120 CALL CLEAR
1130 PRINT "SIE SOLLEN DOCH
EINE ZAHL": :"ZWISCHEN ""1""
 UND ""11""": :"WAEHLEN !":
: : :
1140 GOTO 1000
1150 CALL CLEAR
1160 X=10
1170 Y=2
1180 T$="a = ""3C4299A1A1994
23C"""
1190 GOSUB 2090
1200 GOTO 1810
1210 CALL CLEAR
1220 X=10
1230 Y=2
1240 T$="b = ""3C4299A199423
C"""
1250 GOSUB 2090
1260 GOTO 1810
1270 CALL CLEAR
1280 X=10
1290 Y=2
1300 T$="c = ""FF81B9A1A1B98
1FF"""
1310 GOSUB 2090
1320 GOTO 1810
1330 CALL CLEAR
1340 X=10
1350 Y=2
1360 T$="d = ""FF8199A1A1998
1FF"""
1370 GOSUB 2090
1380 GOTO 1810
1390 CALL CLEAR
1400 X=10
1410 Y=2
1420 T$="e = ""3C42999191994
23C"""
1430 GOSUB 2090
1440 GOTO 1810
1450 CALL CLEAR
1460 X=10
1470 Y=2
1480 T$="f = ""3C42182020184
23C"""
1490 GOSUB 2090
1500 GOTO 1810
1510 CALL CLEAR
1520 X=10
1530 Y=2
1540 T$="g = ""004299A1A1994
2"""
1550 GOSUB 2090
1560 GOTO 1810
1570 CALL CLEAR
```

```
1580 X=10
1590 Y=2
1600 T$="h = ""001C202020201
C"""
1610 GOSUB 2090
1620 GOTO 1810
1630 CALL CLEAR
1640 X=10
1650 Y=2
1660 T$="i = ""003C202020203
C"""
1670 GOSUB 2090
1680 GOTO 1810
1690 CALL CLEAR
1700 X=10
1710 Y=2
1720 T$="j = ""E781B9A1A1B98
1E7"""
1730 GOSUB 2090
1740 GOTO 1810
1750 CALL CLEAR
1760 X=10
1770 Y=2
1780 T$="k = ""E78199A1A1A19
9E7"""
1790 GOSUB 2090
1800 GOTO 1810
1810 X=18
1820 Y=2
1830 T$="WENN FERTIG,"
1840 GOSUB 2090
1850 X=20
1860 Y=2
1870 T$="TASTE DRUECKEN"
1880 GOSUB 2090
1890 CALL KEY(0,K,S)
1900 IF S=0 THEN 1890
1910 CALL HCHAR(16,1,32,320)
1920 X=18
1930 Y=2
1940 T$="MOECHTEN SIE NOCH"
1950 GOSUB 2090
1960 X=20
1970 Y=2
1980 T$="MEHR CODE'S SEHEN ?
"
1990 GOSUB 2090
2000 INPUT " ":N$
2010 IF N$="J" THEN 900
2020 IF N$="N" THEN 2030
2030 CALL CLEAR
2040 PRINT ".....- AUF WIEDE
RSEHEN -": : : : : : : : : :
 : :
2050 FOR I=1 TO 500
2060 NEXT I
2070 PRINT : : : : : : : : : :
 : : : : :
2080 END
2090 FOR W=1 TO LEN(T$)
2100 CALL HCHAR(X,Y+2,ASC(SE
G$(T$,W,1)))
2110 Y=Y+1
2120 NEXT W
2130 RETURN
```

Die nächste TI REVUE erscheint am 28. Oktober

BÖRSE

Verkaufe TI99/4A + 32 k + Centr. + Recorder + Kabel + Ext.-Basic + Literat. + Listings + Programm-Cass. + Mini-roboter mit Steuerungs-programm + 2 Joysticks + Modul „Car Wars"
Tel.: 0781 / 75138

HILFE! Suche ganzes TI-99/4A Gehäuse für ca. 15 bis 20 DM. Angebote an Michael v.d. Mühlen. Tel. 02103/44144

Verkaufe Original TI ext. Basic Modul und Sanyo Monitor mit Kabel (grün/28 cm) im Raum Stuttgart für DM 150,— Bzw. 300,—.
Tel. 0711/748318 nach 18 Uhr.

Verkaufe Attack-Modul für 20 DM !!!
Carsten Simms, Oldauer Heuweg 66,
3107 Hambühren 2, Tel. 05084/3578

Verkaufe Drucker Star DP515 DIN A3, Serielle + Paralelle Schnittstelle. Neupreis 2000
Verk.Preis: 600,— DM
Wie funktioniert Spracheing. MBX ohne Module, wer kann mir helfen? (Gegenleistung)
Tel.: 07175/5031 (9—16 Uhr)

Verk. o. tausche Lern- und Spielmodule (Othello u.a.) u. Apesoft Grafikprog. (mit 40 Grafik-befehlen!) Billig! Tel. 0561/516790

Software-Verkauf für 99/4A: Module, X-Bas., Kass. und Bücher. Tel. 04102-64889

Suche gute Assembler Prg. gegen Tausch. Suche Kontakte mit Akkustikkoppler-User.
Tel. 07175/5031

Tausche TI99/4A-Module Alpiner, Carwars, Othello, TI-Invaders gegen Pal-Modulator f. UHF-Kanal 36 — Tel. 05205/7405

Verkaufe: TI99/4A + EXT.-Basic + Handbücher + Joysticks + Kass. Rec.-Kabel (1) + Literatur + Software, alles original TI! UB 450,— DM
Michael Habermann, Tel.: 06842/514 83 nachmittags.

Tausche od. verkaufe: Cassetten HiFi-Deck NAKAMICHI DRAGON (Referenz bei Stereoplay u. Stereo) m. folgd. Ausstattg. = Revers, Einmesscomputer, autom. Azimuth-Regelung usw. Neupr. DM 4150,— im Tausch gegen kpl. ausgest. TI-99/4A Computer-Anlage od. für DM 3600,—
R. Göhring, Alvesloher Str. 21, 2358 Kaltenkirchen
04191/7305 ab 15.10.85

```
10 REM*********************
11 REM*CASTLE of 81 DOOMS*
12 REM*                 *
13 REM*  Copyright  by   *
14 REM* Gerhard Stegmann *
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*     9070 Bytes   *
25 REM*******************
26 REM
40 CALL CLEAR
50 CALL SCREEN(12)
60 PRINT "*****************
********"
70 PRINT "* THE CASTLE OF 81
 ROOMS *..*(C)BY G.STEGMANN
&......*..*GRAPHIK S.BURKERT
 1/84..*"
80 PRINT "*****************
********"
90 PRINT : : : : : : : : : :
 : : : : : :
100 PRINT "SPIELANLEITUNG?(J
/N)"
110 CALL KEY(0,K,S)
120 IF S=0 THEN 110
130 IF K<>74 THEN 540
140 CALL CLEAR
150 PRINT "SIE BEFINDEN SICH
 IN EINEM":"SCHLOSS MIT 81 R
AEUMEN": :"DIESE RAEUME BEIN
HALTEN"
160 PRINT "VERSCHIEDENE GEGE
NSTAENDE": :"ES KOENNEN JEWE
ILS 2":"GEGENSTAENDE VON EIN
ER ART":"MIT DER TASTE'N'AUF
-"
170 PRINT "GENOMMEN WERDEN":
 :"EIN TIEFER TON NACH DEM "
:"'AKTIONSPIEPS' SIGNALISIER
T"
180 PRINT "DASS NICHTS MEHR
AUFGENOMMEN":"WERDEN KANN."
190 PRINT : :"BITTE EINE TAS
TE DRUECKEN"
200 CALL KEY(0,K,S)
210 IF S=0 THEN 200
220 CALL CLEAR
230 PRINT "DIE BIBEL SCHUETZ
T SIE":"VOR GEISTERN,DIE IHN
EN":"MANCHMAL BEGEGNEN"
240 PRINT "DIE PILLEN SCHUET
ZEN SIE VOR":"DEM'BURGVIRUS'
DER SIE ":"VON ZEIT ZU ZEIT
INFIZIERT"
250 PRINT "KELCHE UND TRUHEN
 KOENNEN ":"SIE SICH NEHMEN,
SOVIEL SIE ":"WOLLEN"
260 PRINT : :"BITTE EINE TAS
TE DRUECKEN"
270 CALL KEY(0,K,S)
280 IF S=0 THEN 270
290 CALL CLEAR
300 PRINT "** DIE PUNKTEVERT
EILUNG **": :
310 PRINT "PUNKTE.I..GEGENST
AND"
320 PRINT "-----------------
---"
330 PRINT ".......I"
340 PRINT "..60...I..TRUHE"
350 PRINT "...2...I..KELCH"
360 PRINT "...2...I..PILLE"
370 PRINT "...2...I..BIBEL"
380 PRINT ".......I"
390 PRINT : :"JEDER SCHRITT
ODER JEDE TAT": :"KOSTET SIE
 2 ENERGIE-":"EINHEITEN": :"
DER KELCH BRINGT 10"
400 PRINT "ENERGIEEINHEITEN
ZURUECK": :"BITTE EINE TASTE
 DRUECKEN"
410 CALL KEY(0,K,S)
420 IF S=0 THEN 410
430 CALL CLEAR
440 PRINT "....** DIE STEUER
UNG **": :
450 PRINT "........V(VORWAER
TS)"
460 PRINT "L(LINKS).........
...R(RECHTS)"
470 PRINT "........Z(ZURUECK
)"
480 PRINT : :"....N(GEGENSTA
ND AUFNEHMEN)": :"DER 'HINWE
IS' ERINNERT":"SIE DARAN,DAS
S DER AUSGANG"
490 PRINT "IN EINEM DER ANGR
ENZENDEN":"RAEUME IST.SIE MU
ESSEN ":"IHN DANN NUR NOCH F
INDEN"
500 PRINT "ALLES KLAR?(J/N)"
510 CALL KEY(0,K,S)
520 IF S=0 THEN 510
530 IF K<>74 THEN 40
540 CALL CLEAR
550 DIM RAU(9,9)
560 ENE=99
570 PU=0
580 REM ** AUFBAU DER RAEUME
 **
590 FOR RAUM=1 TO 9
600 LET ZAHL(RAUM)=0
610 FOR RAUM2=1 TO 9
620 RANDOMIZE
630 LET RAU(RAUM,RAUM2)=INT(
9*RND+1)
640 IF RAU(RAUM,RAUM2)=2 THE
N 620
```

```
650 LET WE(RAUM,RAUM2)=RAU(R
AUM,RAUM2)
660 NEXT RAUM2
670 NEXT RAUM
680 RANDOMIZE
690 HH=INT(9*RND+1)
700 RANDOMIZE
710 BH=INT(9*RND+1)
720 RAU(HH,BH)=2
730 GOSUB 3080
740 GOSUB 2710
750 CALL CLEAR
760 CALL SCREEN(2)
770 FOR C=2 TO 8
780 CALL COLOR(C,16,2)
790 NEXT C
800 FOR Y=1 TO 1500
810 NEXT Y
820 RANDOMIZE
830 HO=INT(9*RND+1)
840 RANDOMIZE
850 BR=INT(9*RND+1)
860 GOSUB 890
870 GOSUB 960
880 GOTO 1340
890 CALL HCHAR(22,1,32,32)
900 CALL SCREEN(2)
910 FOR F=14 TO 16
920 CALL COLOR(F,2,2)
930 NEXT F
940 GOTO 3360
950 RETURN
960 CALL VCHAR(1,4,148,18)
970 CALL VCHAR(1,25,148,18)
980 CALL HCHAR(18,4,148,21)
990 FOR H=1 TO 6
1000 B=4+H
1010 CALL HCHAR(H,B,137)
1020 NEXT H
1030 CALL HCHAR(6,11,139,8)
1040 FOR H=1 TO 6
1050 B=25-H
1060 CALL HCHAR(H,B,136)
1070 NEXT H
1080 CALL VCHAR(7,19,139,9)
1090 CALL VCHAR(7,10,140,9)
1100 CALL VCHAR(9,12,140,6)
1110 CALL VCHAR(9,17,139,6)
1120 CALL HCHAR(8,13,138,4)
1130 CALL HCHAR(15,11,138)
1140 CALL HCHAR(15,18,138)
1150 CALL HCHAR(15,12,150)
1160 CALL HCHAR(15,17,149)
1170 CALL HCHAR(16,19,137)
1180 CALL HCHAR(16,10,136)
1190 CALL HCHAR(16,20,141)
1200 CALL HCHAR(16,9,141)
1210 CALL VCHAR(9,9,143,7)
1220 CALL VCHAR(9,20,143,7)
1230 CALL HCHAR(8,20,142)
1240 CALL HCHAR(8,9,142)
1250 CALL HCHAR(7,9,146)
1260 CALL HCHAR(7,20,144)
1270 CALL HCHAR(7,21,145)
1280 CALL HCHAR(6,21,151)
1290 CALL VCHAR(8,21,140,10)
1300 CALL HCHAR(7,8,147)
1310 CALL HCHAR(6,8,152)
1320 CALL VCHAR(8,8,139,10)
1330 GOSUB 2580
1340 START=RAU(HO,BR)
1350 ON START GOSUB 2020,232
0,2140,2080,2140,2200,2200,2
260,2020
1360 CALL SOUND(100,500,0,60
0,0)
1370 CALL SOUND(100,300,0,40
0,0)
1380 IF WE(HO,BR)<>0 THEN 14
10
1390 WERT=1
1400 G$="NICHTS"
1410 GOSUB 3740
1420 CALL HCHAR(16,17,WERT1)
1430 F$="SIE FINDEN:"
1440 FOR J=1 TO LEN(F$)
1450 CALL HCHAR(20,J+3,ASC(S
EG$(F$,J,1)))
1460 NEXT J
1470 IF HA<>HO THEN 1490
1480 IF BA=BR THEN 2910
1490 FOR J=1 TO LEN(G$)
1500 CALL HCHAR(22,J+3,ASC(S
EG$(G$,J,1)))
1510 NEXT J
1520 CALL KEY(0,K,S)
1530 CALL SOUND(-100,-5,24)
1540 IF S=0 THEN 1520
1550 CALL SOUND(100,500,0,60
0,0)
1560 ST$=CHR$(K)
1570 ENE=ENE-1
1580 IF ENE>0 THEN 1620
1590 CALL CLEAR
1600 PRINT "TUT MIR LEID": :
"SIE HABEN DIE GANZE ENERGIE
 VERBRAUCHT":"PUNKTE:";PU: :
"NOCHMAL?(J)"
1610 GOTO 3490
1620 IF ST$="L" THEN 1700
1630 IF ST$="R" THEN 1730
1640 IF ST$="V" THEN 1760
1650 IF ST$="Z" THEN 1370
1660 IF ST$<>"N" THEN 1520 E
LSE 1790
1670 GOSUB 890
1680 HO=HO+1
1690 GOTO 1330
1700 BR=BR-1
1710 GOSUB 890
1720 GOTO 1330
1730 BR=BR+1
```

```
1740 GOSUB 890
1750 GOTO 1330
1760 HO=HO-1
1770 GOSUB 890
1780 GOTO 1330
1790 IF WERT=1 THEN 1830
1800 IF WERT=6 THEN 1520
1810 IF WE(HO,BR)=0 THEN 183
0
1820 IF ZAHL(WERT)<2 THEN 18
50
1830 CALL SOUND(1000,110,0)
1840 GOTO 1520
1850 IF WERT1=128 THEN 1980
1860 IF WERT1=120 THEN 1950
1870 ZAHL(WERT)=ZAHL(WERT)+1
1880 WE(HO,BR)=0
1890 CALL HCHAR(16,17,32)
1900 FOR Y=600 TO 700 STEP 2
0
1910 CALL SOUND(-100,Y,0,Y+2
0,0)
1920 NEXT Y
1930 PU=PU+2
1940 GOTO 1520
1950 PU=PU+60
1960 WE(HO,BR)=0
1970 GOTO 1890
1980 ENE=ENE+10
1990 WE(HO,BR)=0
2000 PU=PU+2
2010 GOTO 1890
2020 G$="NICHTS"
2030 WERT=1
2040 V=16
2050 H=1
2060 GOSUB 2380
2070 RETURN
2080 G$="EINEN KELCH MIT ENE
RGIE"
2090 WERT=2
2100 V=9
2110 H=15
2120 GOSUB 2380
2130 RETURN
2140 G$="EINE BIBEL GEGEN GE
ISTER"
2150 WERT=3
2160 V=2
2170 H=15
2180 GOSUB 2380
2190 RETURN
2200 G$="EINE PILLE GEGEN IN
FEKTION"
2210 WERT=4
2220 V=8
2230 H=5
2240 GOSUB 2380
2250 RETURN
2260 G$="EINE TRUHE MIT GOLD
"
2270 WERT=5
2280 V=2
2290 H=7
2300 GOSUB 2380
2310 RETURN
2320 G$="EINEN HINWEIS"
2330 WERT=6
2340 V=7
2350 H=12
2360 GOSUB 2380
2370 RETURN
2380 FOR Y=14 TO 16
2390 CALL COLOR(Y,V,H)
2400 NEXT Y
2410 CALL SCREEN(H)
2420 GOSUB 2440
2430 RETURN
2440 EN$="ENERGIE"
2450 FOR E=1 TO LEN(EN$)
2460 CALL HCHAR(19,E+3,ASC(S
EG$(EN$,E,1)))
2470 NEXT E
2480 FOR E=1 TO LEN(STR$(ENE
))
2490 CALL HCHAR(19,11+E,ASC(
SEG$(STR$(ENE),E,1)))
2500 NEXT E
2510 FOR P=1 TO 6
2520 CALL HCHAR(19,16+P,ASC(
SEG$("PUNKTE",P,1)))
2530 NEXT P
2540 FOR P=1 TO LEN(STR$(PU)
)
2550 CALL HCHAR(19,24+P,ASC(
SEG$(STR$(PU),P,1)))
2560 NEXT P
2570 RETURN
2580 IF HO=0 THEN 2630
2590 IF HO=10 THEN 2650
2600 IF BR=10 THEN 2670
2610 IF BR=0 THEN 2690
2620 RETURN
2630 HO=9
2640 RETURN
2650 HO=1
2660 RETURN
2670 BR=1
2680 RETURN
2690 BR=9
2700 RETURN
2710 RANDOMIZE
2720 IF RND>.6 THEN 2820
2730 IF RND>.5 THEN 2780
2740 HA=HH
2750 BA=BH+1
2760 IF BA>9 THEN 2710
2770 RETURN
2780 HA=HH
2790 BA=BH-1
2800 IF BA<1 THEN 2710
2810 RETURN
```

```
2820 IF RND>.8 THEN 2890
2830 HA=HH-1
2840 IF HA<1 THEN 2710
2850 BA=BH
2860 RETURN
2870 HA=HH+1
2880 IF HA>9 THEN 2710
2890 BA=BH
2900 RETURN
2910 CALL CLEAR
2920 CALL SCREEN(15)
2930 PRINT "GRATULIERE!!": :
"DU HAST DEN AUSGANG GEFUNDE
N":"PUNKTE:";PU: :"BONUS:";E
NE*100: :"GESAMT:";PU+ENE*10
0
2940 ZAHL(4)=0
2950 ZAHL(3)=0
2960 RESTORE 3060
2970 FOR SOUND=1 TO 7
2980 READ D,F
2990 CALL SOUND(D,F,0,F+1,0)
3000 NEXT SOUND
3010 PRINT "NOCHMAL?(J)"
3020 CALL KEY(0,K,S)
3030 IF S=0 THEN 3020
3040 IF K=74 THEN 540
3050 END
3060 DATA -100,131,-100,123,
-100,123,-100,165,-100,147,1
00,175,1000,131
3070 REM ** ZEICHENDEFINITIO
N **
3080 CALL CHAR(96,"000000307
87830")
3090 CALL CHAR(104,"7C6C446C
6C6C7C7C")
3100 CALL CHAR(112,"18181818
18001818")
3110 CALL CHAR(120,"00007EFF
81E7FFFF")
3120 CALL CHAR(128,"4444447C
1010107C")
3130 CALL CHAR(136,"01020408
10204080")
3140 CALL CHAR(137,"80402010
08040201")
3150 CALL CHAR(138,"00000000
000000FF")
3160 CALL CHAR(139,"80808080
8080808")
3170 CALL CHAR(140,"01010101
01010101")
3180 CALL CHAR(141,"81818181
818181FF")
3190 CALL CHAR(142,"FF818181
81818181")
3200 CALL CHAR(143,"01010101
81818181")
3210 CALL CHAR(144,"01020408
10204081")
3220 CALL CHAR(145,"03050911
21418101")
3230 CALL CHAR(146,"80402010
08040281")
3240 CALL CHAR(147,"C0A09088
84828180")
3250 CALL CHAR(148,"FFFFFFFF
FFFFFFFF")
3260 CALL CHAR(149,"80808080
808080FF")
3270 CALL CHAR(150,"01010101
010101FF")
3280 CALL CHAR(151,"01030509
11214181")
3290 CALL CHAR(152,"80C0A090
88848281")
3300 CALL COLOR(9,16,1)
3310 CALL COLOR(10,2,1)
3320 CALL COLOR(11,13,1)
3330 CALL COLOR(12,16,1)
3340 CALL COLOR(13,5,1)
3350 RETURN
3360 RANDOMIZE
3370 REM ** GEFAHREN **
3380 IF RND<.79 THEN 950
3390 IF RND>=.8123 THEN 3600
3400 PRINT "IHNEN BEGEGNET E
IN GEIST"
3410 IF ZAHL(3)<1 THEN 3480
3420 PRINT :"ABER SIE HABEN
EINE BIBEL"
3430 FOR H=1 TO 500
3440 NEXT H
3450 CALL CLEAR
3460 ZAHL(3)=ZAHL(3)-1
3470 GOTO 960
3480 PRINT :"UND DA SIE KEIN
E BIBEL":"MEHR HABEN,WERDEN
SIE VON":"IHM GETOETET": :"N
OCHMAL?(J)":"PUNKTE:";PU
3490 RESTORE 3730
3500 FOR TM=1 TO 11
3510 READ DAUER,FREQUENZ
3520 CALL SOUND(DAUER,FREQUE
NZ,0,FREQUENZ+1,0)
3530 NEXT TM
3540 CALL KEY(0,K,S)
3550 IF S=0 THEN 3540
3560 ZAHL(3)=0
3570 ZAHL(4)=0
3580 IF K=74 THEN 540
3590 END
3600 PRINT "SIE WERDEN VON E
INEM VIRUS ":"INFIZIERT"
3610 IF ZAHL(4)<1 THEN 3680
3620 PRINT :"HABEN ABER NOCH
 EINE":"PILLE DIE SIE RETTET
"
3630 ZAHL(4)=ZAHL(4)-1
3640 FOR W=1 TO 500
3650 NEXT W
```

```
3660 CALL CLEAR
3670 GOTO 960
3680 PRINT :"UND WERDEN,DA S
IE KEINE":"PILLE MEHR HABEN,
VON IHM ":"GETOETET":"PUNKTE
:";PU
3690 PRINT : :"NOCHMAL?(J)":
 :
3700 ZAHL(4)=0
3710 ZAHL(3)=0
3720 GOTO 3490
3730 DATA 500,196,500,196,10
0,196,500,196,500,233,100,22
0,250,220,250,196,250,196,25
0,185,1000,196
3740 ON WERT GOTO 3750,3770,
3790,3810,3830,3850
3750 WERT1=32
3760 RETURN
3770 WERT1=128
3780 RETURN
3790 WERT1=104
3800 RETURN
3810 WERT1=96
3820 RETURN
3830 WERT1=120
3840 RETURN
3850 WERT1=112
3860 RETURN
```



```
100 ! ***********************
110 ! *                     *
120 ! *       ECARTE        *
130 ! *    Copyright by     *
140 ! *Christian weichmann  *
150 ! *               ^     *
160 ! * Benoetigte Geraete  *
170 ! *  TI99/4A  Konsole   *
180 ! *     Ext. Basic      *
190 ! *                     *
200 ! *  Speicherbelegung   *
210 ! *     8645 Bytes      *
220 ! ***********************
225 !
230 RANDOMIZE
240 G=INT(RND*2)+1
250 DIM K(2,5),KF(4,8),S(32)
,P(2)
260 CALL VORBEREITUNG(G)
270 KFL=0
280 FOR L1=1 TO 4
290 FOR L2=1 TO 8
300 KF(L1,L2)=-1
310 NEXT L2
320 NEXT L1
330 FOR KT=1 TO 32
340 KN1=INT(RND*4+1)
350 KN2=INT(RND*8+1)
360 IF KF(KN1,KN2)=0 THEN 34
0
370 S(KT)=10*KN1+KN2
380 KF(KN1,KN2)=0
390 NEXT KT
400 Z=32
410 FOR KT=1 TO 5
420 FOR SP=3-G TO G STEP ((G
<3-G)+.5)*2
430 K(SP,KT)=S(Z)
440 Z=Z-1
450 CALL KARTE(-(SP=1)*S(Z+1
),5*(KT-1)+1,5+6*SP)
460 NEXT SP
470 NEXT KT
480 TK=S(Z)
490 T=INT(TK/10)
500 IF TK-T*10=8 THEN P(G)=P
(G)+1 :: KFL=-1 :: CALL BILD
SCHIRM(0,P(),0,0,G)
510 CALL KARTE(TK,3,14)
520 CALL KARTE(0,8,14)
530 FOR SZ=1 TO 21
540 IF INT(S(SZ)/10)=T THEN
S(SZ)=S(SZ)+100
550 NEXT SZ
560 AZ=32
570 SZ=21
580 FOR SP=1 TO 2
590 FOR I=1 TO 5
600 IF T=INT(K(SP,I)/10)THEN
 K(SP,I)=K(SP,I)+100
610 NEXT I
```

```
620 NEXT SP
630 CALL SORTIER(K(,))
640 FOR I=1 TO 5
650 CALL KARTE(K(1,I),(I-1)*
5+1,11)
660 NEXT I
670 M,STZ(1),STZ(2)=0
680 IF G=1 THEN CALL SPIEL(T
,P(),K(,),KFL,M):: M=M+(M=-1
):: CALL BILDSCHIRM(0,P(),0,
1-M,G)
690 IF M=-3 THEN GOTO 680
700 IF M THEN SP=2 :: GOTO 9
30
710 CALL KEY(0,M,Z):: IF Z=0
 THEN 710
720 IF M=65 AND K(1,1)=108+T
*10 AND NOT KFL THEN P(1)=P(
1)+1 :: KFL=NOT KFL :: CALL
BILDSCHIRM(0,P(),4,-1,G):: G
OTO 710
730 IF M=83 THEN SP=1 :: CAL
L BILDSCHIRM(0,P(),3+(G=1),-
1,G):: GOTO 930
740 IF M<>75 THEN 710
750 IF G=2 THEN CALL BILDSCH
IRM(0,P(),1,-1,G):: CALL SPI
EL(T,P(),K(,),KFL,M):: IF M
THEN SP=2 :: CALL BILDSCHIRM
(0,P(),-1,2,G):: GOTO 930
760 DISPLAY AT(12,(G-1)*21+1
)SIZE(7):"WIEVIEL"
770 IF G=2 THEN CALL KEY(0,A
,ST):: IF ST<1 OR A<48 OR A>
53 THEN 770 ELSE A=VAL(CHR$(
A)):: GOTO 790
780 CALL AUSTAUSCH(K(,),A)
790 DISPLAY AT(12,(G-2)*-21+
1)SIZE(7):A
800 ON G GOSUB 840,880
810 IF G=1 THEN CALL KEY(0,A
,ST):: IF ST<1 OR A<48 OR A>
53 THEN 810 ELSE A=VAL(CHR$(
A))
820 ON G GOSUB 880,840
830 CALL VCHAR(1,11,31,24)::
 CALL KARTE(0,18,14):: GOTO
630
840 FOR L=1 TO 5
850 IF SZ=0 THEN CALL KARTE(
-1,8,14):: RETURN
860 IF K(2,L)<100 THEN S(AZ)
=K(2,L):: AZ=AZ-1 :: K(2,L)=
S(SZ):: SZ=SZ-1
870 NEXT L :: RETURN
880 FOR L=1 TO A
890 IF SZ=0 THEN CALL KARTE(
-1,8,14):: RETURN
900 CALL SOUND(100,110,1)
910 CALL KEY(0,A,ST):: IF ST
<1 OR A<49 OR A>53 THEN 910
ELSE A=VAL(CHR$(A)):: S(AZ)=
K(1,A):: AZ=AZ-1 :: K(1,A)=S
(SZ):: SZ=SZ-1 :: CALL HCHAR
((A-1)*5+1,11,62)
920 NEXT L :: RETURN
930 A=(NOT(G-2))+2
940 IF A=2 THEN CALL AUSSPIE
L1(K(,),ASK):: GOTO 1050
950 CALL SOUND(100,110,1)
960 CALL KEY(0,KT,ST):: IF S
T=0 THEN 960
970 IF KT<49 OR KT>53 THEN 9
50
980 KT=VAL(CHR$(KT))
990 IF K(1,KT)=-1 THEN 950
1000 CALL KARTE(-1,5*(KT-1)+
1,11)
1010 ASK=K(1,KT)
1020 CALL KARTE(ASK,13,14)
1030 K(1,KT)=-1
1040 CALL AUSSPIEL2(T,K(,),A
SK,BGK):: GOTO 1150
1050 CALL SOUND(100,110,1)
1060 CALL KEY(0,KT,ST):: IF
ST=0 THEN 1060
1070 IF KT<49 OR KT>53 THEN
1050
1080 KT=VAL(CHR$(KT))
1090 IF K(1,KT)=-1 THEN 1050
1100 CALL KONTROLLIER(T,K(,)
,ASK,KT)
1110 CALL KARTE(-1,5*(KT-1)+
1,11)
1120 BGK=K(1,KT)
1130 CALL KARTE(BGK,13,14)
1140 K(1,KT)=-1
1150 IF INT(BGK/10)<>INT(ASK
/10)AND INT(BGK/10)<>10+T TH
EN BGK=-BGK
1160 A=-(ASK>BGK)*A-((NOT(A-
2))+2)*(ASK<BGK)
1170 STZ(A)=STZ(A)+1
1180 CALL KARTE(-1,13,14)
1190 CALL KARTE(0,5*(STZ(A)-
1)+1,A*12-4)
1200 IF STZ(1)+STZ(2)<5 THEN
 940
1210 IF STZ(SP)>2 THEN P(SP)
=P(SP)+1
1220 IF STZ(SP)=5 THEN P(SP)
=P(SP)+1
1230 IF STZ(SP)<3 THEN P(SP)
=P(SP)-2 :: SP=(NOT(SP-2))+2
1240 DISPLAY AT(20,(SP-1)*21
+1)SIZE(7):"SIEGER"
1250 CALL BILDSCHIRM(0,P(),0
,0,G)
1260 DISPLAY AT(21,22)SIZE(7
):"NOCH"
1270 DISPLAY AT(22,22)SIZE(7
):"EIN"
```

```
1280 DISPLAY AT(23,22)SIZE(7
):"SPIEL?"
1290 CALL KEY(Ø,A,ST):: IF S
T=Ø THEN 129Ø
13ØØ IF A=78 THEN STOP
131Ø IF A<>74 THEN 128Ø
132Ø G=(NOT(G-2))+2
133Ø CALL BILDSCHIRM(-1,P(),
Ø,Ø,G)
134Ø GOTO 27Ø
135Ø SUB AUSTAUSCH(K(,),A)
136Ø A=Ø :: FOR L=1 TO 5
137Ø IF K(2,L)<1ØØ THEN A=A+
1
138Ø NEXT L :: SUBEND
139Ø SUB AUSSPIEL1(K(,),ASK)
14ØØ M=-1 :: KT=Ø
141Ø FOR L=1 TO 5
142Ø IF K(2,L)>M THEN M=K(2,
L):: KT=L
143Ø NEXT L
144Ø CALL KARTE(-1,(KT-1)*5+
1,17)
145Ø CALL KARTE(K(2,KT),13,1
4)
146Ø ASK=K(2,KT)
147Ø K(2,KT)=-1
148Ø SUBEND
149Ø SUB AUSSPIEL2(T,K(,),AS
K,BGK)
15ØØ M=2ØØ :: KT=Ø
151Ø FOR L=1 TO 5
152Ø IF K(2,L)<M AND K(2,L)>
ASK AND INT(K(2,L)/1Ø)=INT(A
SK/1Ø)THEN M=K(2,L):: KT=L
153Ø NEXT L
154Ø IF KT=Ø THEN 16ØØ
155Ø CALL KARTE(-1,(KT-1)*5+
1,17)
156Ø CALL KARTE(K(2,KT),13,1
4)
157Ø BGK=K(2,KT)
158Ø K(2,KT)=-1
159Ø SUBEXIT
16ØØ FOR L=1 TO 5
161Ø IF M>K(2,L)AND INT(K(2,
L)/1Ø)=INT(ASK/1Ø)THEN M=K(2
,L):: KT=L
162Ø NEXT L
163Ø IF KT>Ø THEN 155Ø
164Ø FOR L=1 TO 5
165Ø IF K(2,L)<M AND INT(K(2
,L)/1Ø)=1Ø+T THEN M=K(2,L)::
 KT=L
166Ø NEXT L
167Ø IF KT>Ø THEN 155Ø
168Ø M=9
169Ø FOR L=1 TO 5
17ØØ IF M-INT(M/1Ø)*1Ø>K(2,L
)-INT(K(2,L)/1Ø)*1Ø AND K(2,
L)>-1 THEN M=K(2,L):: KT=L
171Ø NEXT L :: GOTO 155Ø
172Ø SUBEND
173Ø SUB KONTROLLIER(T,K(,),
ASK,KTA)
174Ø BGK=K(1,KTA)
175Ø IF BGK>ASK AND INT(BGK/
1Ø)=INT(ASK/1Ø)THEN 191Ø
176Ø M=2ØØ :: KT=Ø :: FOR L=
1 TO 5
177Ø IF INT(K(1,L)/1Ø)=INT(A
SK/1Ø)AND K(1,L)<M AND K(1,L
)>ASK THEN KT=L :: M=K(1,L)
178Ø NEXT L
179Ø IF KT<>Ø THEN 19ØØ
18ØØ IF INT(BGK/1Ø)=INT(ASK/
1Ø)THEN 191Ø
181Ø FOR L=1 TO 5
182Ø IF INT(K(1,L)/1Ø)=INT(A
SK/1Ø)AND K(1,L)<M THEN M=K(
1,L):: KT=L
183Ø NEXT L
184Ø IF KT<>Ø THEN 19ØØ
185Ø IF INT(BGK/1Ø)=1Ø+T THE
N 191Ø
186Ø FOR L=1 TO 5
187Ø IF M>K(1,L)AND INT(K(1,
L)/1Ø)=1Ø+T THEN M=K(1,L)::
KT=L
188Ø NEXT L
189Ø IF KT=Ø THEN 191Ø
19ØØ KTA=KT
191Ø SUBEND
192Ø SUB VORBEREITUNG(G)
193Ø FOR SZ=95 TO 143
194Ø READ S$
195Ø CALL CHAR(SZ,S$)
196Ø NEXT SZ
197Ø P(1),P(2)=Ø
198Ø CALL BILDSCHIRM(-1,P(),
Ø,Ø,G)
199Ø FOR I=8 TO 14
2ØØØ CALL COLOR(I,2,16)
2Ø1Ø NEXT I
2Ø2Ø CALL COLOR(13,7,16)
2Ø3Ø CALL COLOR(Ø,1,13)
2Ø4Ø DATA AA55AA55AA55AA55,Ø
Ø3F3FØØØØØØØØ3,Ø3ØFØFØFØFØF
ØFØØ,ØØFØFØ3Ø3ØCØCØØØ,ØØ,ØØØ
FØF3Ø3Ø3ØØF,ØF3Ø3Ø3ØØFØF
ØØ
2Ø5Ø DATA ØØCØCØ3Ø3Ø3ØCØ,C
Ø3Ø3Ø3ØCØCØØØ,ØØØFØF3Ø3Ø3Ø
3ØØF,ØFØØØØØØØØFØFØØ,ØØCØCØ
3Ø3Ø3ØFØ,FØ3Ø3ØCØCØØØØØØ
2Ø6Ø DATA ØØ3F3FØØØØØØØØ3,Ø
3ØCØC3Ø3Ø3F3FØØ,ØØFØFØ3Ø3ØCØ
CØØØ,ØØØØØØØØØFØFØØØ,ØØØFØF
3Ø3Ø3Ø3F,3F3Ø3Ø3Ø3Ø3ØØØ
2Ø7Ø DATA ØØCØCØ3Ø3Ø3ØFØ,F
Ø3Ø3Ø3Ø3Ø3Ø3ØØØ,ØØ3F3FØCØCØC
ØCØF,ØFØCØCØCØC3F3FØØ,ØØCØCØ
```

```
30303030C0,C0303030300C0C000
2080 DATA 003F3F0C0C0C0C0C,0
C0C0C0C0C3F3F00,00C0C0303030
3030,303030303030C0C000,003030
303033333C,3C33333030303000
2090 DATA 003030C0C0000000,0
00000C0C0303000,000103070F1F
3F7F,7F3F1F0F07030100,0080C0
E0F0F8FCFE,FEFCF8F0E0C08000
2100 DATA 000000183C3E3F3F,1
F0F0703010000000,000000183C7C
FCFC,F8F0E0C080000000,000103
070F1F3F3F,3F3D190101010300
2110 DATA 0080C0E0F0F8FCFC,F
CBC9880808 0C000,000103070311
397F,7F3911010101 0300,0080C0
E0C0889CFE,FE9C8880 8080C000
2120 SUBEND
2130 SUB BILDSCHIRM(CF,P(),K
1,K2,G)
2140 IF CF THEN CALL CLEAR :
: CALL VCHAR(1,10,31,336)
2150 DISPLAY AT(1,1)SIZE(7):
"SPIELER"
2160 DISPLAY AT(1,22)SIZE(7)
:"SPIELER"
2170 DISPLAY AT(2,1)SIZE(2):
"#1"
2180 DISPLAY AT(2,22)SIZE(2)
:"#2"
2190 DISPLAY AT(4,1)SIZE(7):
"PUNKTE:"
2200 DISPLAY AT(4,22)SIZE(7)
:"PUNKTE:"
2210 DISPLAY AT(5,1)SIZE(7):
P(1)
2220 DISPLAY AT(5,22)SIZE(7)
:P(2)
2230 IF G=1 THEN DISPLAY AT(
7,1)SIZE(5):"GEBER" :: DISPL
AY AT(7,22)SIZE(7):"AUSSP."
2240 IF G=2 THEN DISPLAY AT(
7,22)SIZE(5):"GEBER" :: DISP
LAY AT(7,1)SIZE(7):"AUSSP."
2250 IF K1=1 OR K2=1 THEN DI
SPLAY AT(9,ABS((K1=1)+(K2=1)
*22))SIZE(7):"KARTE"
2260 IF K1=2 OR K2=2 THEN DI
SPLAY AT(9,ABS((K1=2)+(K2=2)
*22))SIZE(7):"SPIEL!"
2270 IF K1=3 OR K2=3 THEN DI
SPLAY AT(9,ABS((K1=3)+(K2=3)
*22))SIZE(7):"ICH" :: DISPLA
Y AT(10,ABS((K1=3)+(K2=3)*22
))SIZE(7):"SPIELE"
2280 IF K1=4 OR K2=4 THEN DI
SPLAY AT(9,ABS((K1=4)+(K2=4)
*22))SIZE(7):"MELDE" :: DISP
LAY AT(10,ABS((K1=4)+(K2=4)*
22))SIZE(7):"KOENIG!"
2290 IF K1=0 THEN DISPLAY AT
(9,1)SIZE(7):"" :: DISPLAY A
T(10,1)SIZE(7):""
2300 IF K2=0 THEN DISPLAY AT
(9,22)SIZE(7):"" :: DISPLAY
AT(10,22)SIZE(7):""
2310 SUBEND
2320 SUB KARTE(C,Z,S)
2330 S=S+2
2340 IF C<=0 THEN 2480
2350 FC=INT(C/10)-INT(C/100)
*10
2360 KC=C-FC*10+100*(C>100)
2370 FC=124+FC*4
2380 KC=92+KC*4
2390 CALL HCHAR(Z,S,FC)
2400 CALL HCHAR(Z,S+1,FC+2)
2410 CALL HCHAR(Z+1,S,FC+1)
2420 CALL HCHAR(Z+1,S+1,FC+3
)
2430 CALL HCHAR(Z+2,S,KC)
2440 CALL HCHAR(Z+2,S+1,KC+2
)
2450 CALL HCHAR(Z+3,S,KC+1)
2460 CALL HCHAR(Z+3,S+1,KC+3
)
2470 SUBEXIT
2480 IF C=0 THEN C=95 ELSE C
=31
2490 CALL VCHAR(Z,S,C,4)
2500 CALL VCHAR(Z,S+1,C,4)
2510 SUBEND
2520 SUB SORTIER(K(,))
2530 FOR I1=1 TO 4
2540 FOR I2=I1+1 TO 5
2550 IF K(1,I1)<K(1,I2)THEN
H=K(1,I2):: K(1,I2)=K(1,I1):
: K(1,I1)=H
2560 NEXT I2
2570 NEXT I1
2580 SUBEND
2590 SUB SPIEL(T,P(),K(,),KF
L,M)
2600 TZ,KZ,DZ,BZ=0
2610 FOR L=1 TO 5
2620 IF K(2,L)=108+T*10 AND(
NOT(KF2))THEN 2740
2630 KI=INT(K(2,L)/10)
2640 IF KI=10+T THEN TZ=TZ+1
 :: GOTO 2680
2650 IF K(2,L)-KI*10=8 THEN
KZ=KZ+1 :: GOTO 2680
2660 IF K(2,L)-KI*10=7 THEN
DZ=DZ+1 :: GOTO 2680
2670 IF K(2,L)-KI*10=6 THEN
BZ=BZ+1 :: GOTO 2680
2680 NEXT L
2690 IF TZ>=3 THEN M=-1 :: G
OTO 2730
2700 IF TZ>=2 AND KZ=2 THEN
M=-1 :: GOTO 2730
2710 IF KZ+DZ+BZ>=4 AND KZ+D
```

```
Z)>=2 THEN M=-1 :: GOTO 2730
2720 M=0
2730 SUBEXIT
2740 IF KFL THEN P(1)=P(1)-1
2750 P(2)=P(2)+1
2760 M=-3
2770 KF2,KFL=-1
2780 SUBEND
```

BÖRSE


```
10 REM*******************
11 REM*                 *
12 REM* CATCH WITH CAR  *
13 REM*  Copyright by   *
14 REM* Dirk  Junghans  *
15 REM*                 *
16 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*    6762 Bytes   *
25 REM*******************
26 REM
160 RANDOMIZE
170 REM DEFINE CHARACTERS
180 DATA 0003060606060606,04
CA4440206E8A8A,000000F08888E
424,7C80C060301B0F01,EAEA2A2
A2A2A2EA1,2464C48282828408
190 DATA 01,9ECC6071311C0F03
,304080000080
200 CALL CHAR(136,"427E5A181
85A7E42")
210 CALL CHAR(137,"00E7427E7
E42E700")
220 CALL CHAR(96,"02C2241818
244340")
230 CALL CHAR(97,"C324187E5A
663C42")
240 CALL CHAR(33,"3C18A5DBDB
A5183C")
250 CALL CHAR(104,"566A566A5
66A566A")
260 CALL CHAR(105,"00FFAA55A
A55FF00")
270 CALL CHAR(112,"18183CE7E
73C1818")
280 CALL CHAR(106,"808080808
0808080")
290 REM VORSPANN
300 RESTORE 180
310 FOR I=144 TO 152
320 READ A$
330 CALL CHAR(I,A$)
340 NEXT I
350 CALL CLEAR
360 CALL SCREEN(10)
370 PRINT TAB(8);"DIRK JUNGH
ANS ": : :TAB(9);"PRAESENTIE
RT":TAB(11);"FUER DEN"
380 PRINT : : :TAB(11);"TI99
/4A": :TAB(13);"VON": : : :
390 PRINT TAB(13);CHR$(144);
CHR$(145);CHR$(146):TAB(13);
CHR$(147);CHR$(148);CHR$(149
)
400 PRINT TAB(13);CHR$(150);
CHR$(151);CHR$(152): : : :
410 CALL SOUND(300,262,2)
420 CALL SOUND(900,349,2)
```

```
430 CALL SOUND(100,329,2)
440 CALL SOUND(100,349,2)
450 CALL SOUND(100,392,2)
460 CALL SOUND(300,440,2)
470 CALL SOUND(300,440,2)
480 CALL SOUND(300,440,2)
490 CALL SOUND(100,440,2)
500 CALL SOUND(100,466,2)
510 CALL SOUND(100,392,2)
520 CALL SOUND(450,440,2)
530 CALL SOUND(150,392,2)
540 CALL SOUND(300,349,2)
550 FOR I=1 TO 500
560 NEXT I
570 FOR I=1 TO 20
580 PRINT
590 NEXT I
600 CALL SCREEN(11)
610 Z=10
620 S=2
630 Q$="*******CATCH WITH CA
R*********"
640 GOSUB 2810
650 FOR I=1 TO 30
660 IF I>13 THEN 700
670 CALL SCREEN(I+1)
680 CALL SCREEN(I+3)
690 CALL SCREEN(I+2)
700 CALL SOUND(100,-7,I,440,
I,190,I,880,I)
710 NEXT I
720 CALL CLEAR
730 WAGEN=3
740 PUNKTE=100
750 R=1
760 ANJ=0
770 DD=0
780 CALL SCREEN(13)
790 CALL COLOR(1,11,15)
800 CALL COLOR(9,10,15)
810 CALL COLOR(10,2,15)
820 CALL COLOR(11,8,15)
830 CALL COLOR(14,6,15)
840 CALL VCHAR(1,31,106,24)
850 Q$="PUNKTE"
860 Z=3
870 S=32
880 GOSUB 2850
890 GOSUB 2890
900 CALL VCHAR(19,31,137,WAG
EN)
910 D=24
920 J=15
930 CALL HCHAR(D,J,136)
940 CALL KEY(2,K,ST)
950 IF ST=0 THEN 940
960 GOSUB 2690
970 CALL JOYST(2,K,ST)
980 IF K>0 THEN 1310
990 IF K<0 THEN 1450
1000 IF ST>0 THEN 1030
1010 IF ST<0 THEN 1170
1020 ON R GOTO 1030,1170,131
0,1450
1030 R=1
1040 L=136
1050 M=D
1060 N=J
1070 O=104
1080 D=D-1
1090 IF D<1 THEN 1590
1100 CALL GCHAR(D,J,U)
1110 IF U=32 THEN 1140
1120 GOSUB 3310
1130 GOTO 970
1140 CALL HCHAR(D,J,136)
1150 CALL HCHAR(D+1,J,104)
1160 GOTO 970
1170 R=2
1180 L=136
1190 M=D
1200 N=J
1210 O=104
1220 D=D+1
1230 IF D>24 THEN 1590
1240 CALL GCHAR(D,J,U)
1250 IF U=32 THEN 1280
1260 GOSUB 3310
1270 GOTO 970
1280 CALL HCHAR(D,J,136)
1290 CALL HCHAR(D-1,J,104)
1300 GOTO 970
1310 R=3
1320 L=137
1330 M=D
1340 N=J
1350 O=105
1360 J=J+1
1370 IF J>30 THEN 1590
1380 CALL GCHAR(D,J,U)
1390 IF U=32 THEN 1420
1400 GOSUB 3310
1410 GOTO 970
1420 CALL HCHAR(D,J,137)
1430 CALL HCHAR(D,J-1,105)
1440 GOTO 970
1450 R=4
1460 L=137
1470 M=D
1480 N=J
1490 O=105
1500 J=J-1
1510 IF J<1 THEN 1590
1520 CALL GCHAR(D,J,U)
1530 IF U=32 THEN 1560
1540 GOSUB 3310
1550 GOTO 970
1560 CALL HCHAR(D,J,137)
1570 CALL HCHAR(D,J+1,105)
1580 GOTO 970
1590 FOR I=1 TO 30
1600 CALL SOUND(-65,119,I,-5
```

```
,I,117,I,115,I)
1610 NEXT I
1620 GOSUB 3000
1630 IF D<1 THEN 1670
1640 IF D>24 THEN 1690
1650 IF J<1 THEN 1710
1660 IF J>30 THEN 1730
1670 D=1
1680 GOTO 1740
1690 D=24
1700 GOTO 1740
1710 J=1
1720 GOTO 1740
1730 J=30
1740 CALL HCHAR(D,J,32)
1750 D=INT(RND*24)+1
1760 J=INT(RND*30)+1
1770 CALL GCHAR(D,J,ZZ)
1780 IF ZZ<>32 THEN 1750
1790 IF J<25 THEN 1800 ELSE
1750
1800 R=3
1810 CALL HCHAR(D,J,137)
1820 GOTO 940
1830 CALL HCHAR(D,J,L)
1840 CALL HCHAR(M,N,O)
1850 CALL SOUND(100,440,1,55
0,1,660,1)
1860 CALL SOUND(100,990,1,10
00,1,1010,1)
1870 PUNKTE=PUNKTE+211
1880 GOSUB 2890
1890 GOTO 2690
1900 CALL HCHAR(D,J,L)
1910 CALL HCHAR(M,N,O)
1920 TON=220
1930 FOR I=1 TO 3
1940 CALL SOUND(100,TON,1)
1950 TON=TON+1
1960 NEXT I
1970 PUNKTE=PUNKTE+311
1980 GOSUB 2890
1990 GOTO 2690
2000 CALL HCHAR(D,J,L)
2010 CALL HCHAR(M,N,O)
2020 CALL SOUND(100,220,1)
2030 CALL SOUND(100,247,1)
2040 CALL SOUND(100,294,1)
2050 CALL SOUND(100,330,1)
2060 FOR I=1 TO 6
2070 DJ1=DJ1+1
2080 IF DJ1>7 THEN 2150
2090 X=INT(RND*24)+1
2100 Y=INT(RND*30)+1
2110 CALL GCHAR(X,Y,F)
2120 IF (F=105)+(F=104)THEN
2130 ELSE 2070
2130 CALL HCHAR(X,Y,32)
2140 CALL SOUND(100,440,1)
2150 DJ1=0
2160 NEXT I
2170 GOTO 2690
2180 CALL HCHAR(D,J,L)
2190 CALL HCHAR(M,N,O)
2200 CALL SOUND(100,262,1)
2210 CALL SOUND(100,524,1)
2220 CALL SOUND(100,392,1)
2230 CALL SOUND(100,440,1)
2240 CALL SOUND(100,344,1)
2250 PUNKTE=PUNKTE+90
2260 GOSUB 2890
2270 FOR I=1 TO 3
2280 DJ=DJ+1
2290 IF DJ>14 THEN 2360
2300 X=INT(RND*24)+1
2310 Y=INT(RND*30)+1
2320 CALL GCHAR(X,Y,F)
2330 IF (F=104)+(F=105)THEN
2340 ELSE 2280
2340 CALL HCHAR(X,Y,32)
2350 CALL SOUND(100,440,1)
2360 DJ=0
2370 NEXT I
2380 GOTO 2690
2390 CALL HCHAR(D,J,L)
2400 CALL HCHAR(M,N,O)
2410 CALL SOUND(100,-6,1,200
,1)
2420 ANJ=ANJ+1
2430 IF ANJ>15 THEN 3040
2440 FOR I=1 TO LEN(STR$(ANJ
))
2450 CALL HCHAR(1,30+I,ASC(S
EG$(STR$(ANJ),I,1)))
2460 NEXT I
2470 PUNKTE=PUNKTE-100
2480 GOSUB 2890
2490 GOTO 970
2500 DD=DD+1
2510 IF DD>1 THEN 2950
2520 RESTORE 2670
2530 FOR I=1 TO 13
2540 READ AB,CD
2550 CALL SOUND(AB,CD,1)
2560 NEXT I
2570 WAGEN=WAGEN+1
2580 CALL VCHAR(18+WAGEN,31,
137)
2590 FOR I=1 TO 25
2600 X=INT(RND*24)+1
2610 Y=INT(RND*30)+1
2620 CALL GCHAR(X,Y,U)
2630 IF (U=104)+(U=105)THEN
2640 ELSE 2600
2640 CALL HCHAR(X,Y,32)
2650 NEXT I
2660 GOTO 2950
2670 DATA 200,330,200,330,20
0,330,200,349,200,262,200,33
0,500,294,200,311,200,311,20
0,311,200,349,300,277
2680 DATA 500,247
```

```
2690 Z$="'a!p"
2700 Z=INT(RND*4)+1
2710 V=ASC(SEG$(Z$,Z,1))
2720 C=INT(RND*24)+1
2730 G=INT(RND*30)+2
2740 CALL GCHAR(C,G,E)
2750 IF E<>32 THEN 2720
2760 CALL SOUND(100,330,1)
2770 CALL SOUND(100,330,1)
2780 CALL SOUND(210,500,1)
2790 CALL HCHAR(C,G,V)
2800 RETURN
2810 FOR I=1 TO LEN(Q$)
2820 CALL HCHAR(Z,S+I,ASC(SE
G$(Q$,I,1)))
2830 NEXT I
2840 RETURN
2850 FOR I=1 TO LEN(Q$)
2860 CALL VCHAR(Z+I,S,ASC(SE
G$(Q$,I,1)))
2870 NEXT I
2880 RETURN
2890 IF PUNKTE<=0 THEN 2900
ELSE 2940
2900 PUNKTE=0
2910 CALL VCHAR(10,32,32,5)
2920 CALL VCHAR(10,32,48)
2930 GOTO 3000
2940 IF PUNKTE>1500 THEN 250
0
2950 CALL VCHAR(10,32,32,5)
2960 FOR I=1 TO LEN(STR$(PUN
KTE))
2970 CALL VCHAR(10+I,32,ASC(
SEG$(STR$(PUNKTE),I,1)))
2980 NEXT I
2990 RETURN
3000 WAGEN=WAGEN-1
3010 IF WAGEN<=0 THEN 3040
3020 CALL VCHAR(19+WAGEN,31,
32)
3030 RETURN
3040 CALL HCHAR(10,12,32,9)
3050 Q$="GAME OVER"
3060 Z=10
3070 S=12
3080 GOSUB 2810
3090 IF PUNKTE>HIGH THEN 311
0
3100 GOTO 3170
3110 HIGH=PUNKTE
3120 TON=110
3130 FOR I=1 TO 3
3140 CALL SOUND(100,TON,1)
3150 TON=TON+110
3160 NEXT I
3170 CALL HCHAR(15,5,32,14)
3180 Z=15
3190 S=5
3200 Q$="HIGHSCORE "&STR$(HI
GH)
3210 GOSUB 2810
3220 CALL HCHAR(20,5,32,18)
3230 Z=20
3240 S=5
3250 Q$="NEUES SPIEL  (J/N)"
3260 GOSUB 2810
3270 CALL KEY(3,K,ST)
3280 IF K=74 THEN 290
3290 IF K=78 THEN 3300 ELSE
3270
3300 STOP
3310 IF U=96 THEN 1830
3320 IF U=97 THEN 1900
3330 IF U=33 THEN 2000
3340 IF U=112 THEN 2180
3350 IF U=104 THEN 2390
3360 IF U=105 THEN 2390
3370 END
```

Die nächste
TI REVUE erscheint
am 28. Oktober

```
10 REM*******************
11 REM*                 *
12 REM*    SKYDRIVER    *
13 REM*  Copyright  by  *
14 REM*   B.Jaehrling   *
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM* Speicherbelegung *
24 REM*    9871 Bytes    *
25 REM*******************
26 REM
160 CALL CLEAR
170 CALL SCREEN(4)
180 REM ZEICHENDEFINITION
190 REM  WOLKEN
200 CALL CHAR(40,"000006070F
3B7F77")
210 CALL CHAR(41,"0078B052CE
FFF6DC")
220 CALL CHAR(42,"663D0C")
230 CALL CHAR(43,"70C87060")
240 CALL CHAR(152,"000000000
3020F07")
250 CALL CHAR(153,"0102386E3
F7FFF7F")
260 CALL CHAR(154,"0461F86BF
DFFBFFF")
270 CALL CHAR(155,"80E8FC914
0D8E87C")
280 CALL CHAR(156,"071F7BEE7
D0C02")
290 CALL CHAR(157,"F7B3FF5F5
3E17706")
300 CALL CHAR(158,"FFFFDFFF9
EDA84")
310 CALL CHAR(159,"FAB8ECB6E
641E420")
320 REM BAUEME
330 CALL CHAR(136,"1818183C7
EBD3C7E")
340 CALL CHAR(137,"000101030
6000103")
350 CALL CHAR(138,"FFFFFFFFF
FFFFFFF")
360 CALL CHAR(139,"008080C06
00080C0")
370 CALL CHAR(140,"03170E000
30F1F33")
380 CALL CHAR(141,"C0E87000C
0F0F8CC")
390 CALL CHAR(142,"27070F0F1
F3FFF7F")
400 CALL CHAR(143,"E4E0F0F0F
8FCFFFE")
410 CALL CHAR(144,"3C3C3C3C3
C3C7EFF")
420 REM  HUBSCHRAUBER
430 CALL CHAR(130,"000005060
F1703")
440 CALL CHAR(131,"00BF0003F
FFFFF07")
450 CALL CHAR(132,"00FE80E01
00888F0")
460 CALL CHAR(133,"030907")
470 CALL CHAR(134,"E048F0")
480 REM LANDEMARKE
490 CALL CHAR(112,"7FFFFF7F"
)
500 CALL CHAR(113,"FFFFFFFF"
)
510 CALL CHAR(115,"FEFFFFFE"
)
520 REM SPRINGER
530 CALL CHAR(120,"18187EBD3
C244281")
540 CALL CHAR(122,"3C7EFFC34
22418")
550 FOR I=3 TO 8
560 CALL COLOR(I,2,4)
570 NEXT I
580 PRINT TAB(7);"S K Y - D
I V E R": : : : :
590 PRINT TAB(14);"von": : :
 : :
600 PRINT TAB(7);"Burkhard J
aehrling": : : : : : : :
610 PRINT TAB(16);"Erkrath,
1984": :
620 CALL HCHAR(6,9,61,17)
630 DATA 300,350,200,330,200
,350,200,370,200,370,200,350
,200,370,200,390
640 DATA 200,410,350,390,200
,370,200,370,450,390
650 FOR K=1 TO 2
660 FOR I=1 TO 13
670 READ A,B
680 CALL SOUND(A-20,B,0,B-2,
5,B+2,5)
690 NEXT I
700 FOR L=1 TO 70
710 NEXT L
720 RESTORE 630
730 NEXT K
740 DATA 300,390,200,390,200
,350,200,350,200,290,200,290
,200,250,200,310,200,280,600
,250,200,200
750 RESTORE 740
760 FOR I=1 TO 10
770 READ A,B
780 CALL SOUND(A-30,B,0,B-2,
5,B+2,5)
790 NEXT I
800 RESTORE 630
810 FOR I=1 TO 23
820 READ A,B
830 CALL SOUND(A-30,B,0,B-2,
5,B+2,5)
840 NEXT I
```

```
850 CALL CLEAR
860 PRINT "   KURZE SPIELANL
EITUNG ?": : :
870 PRINT TAB(12);"[J;N]": :
 : : : : : : : :
880 CALL KEY(0,K,S)
890 IF S=0 THEN 880
900 IF K=78 THEN 930
910 IF K<>74 THEN 880
920 GOSUB 4060
930 CALL CLEAR
940 SP$="BITTE SPIELART WAEH
LEN:"
950 SA$="1; LEICHTERES SPIEL
"
960 SB$="2; ETWAS SCHWIERIGE
R"
970 FOR I=1 TO LEN(SP$)
980 CALL HCHAR(12,4+I,ASC(SE
G$(SP$,I,1)))
990 NEXT I
1000 FOR I=1 TO LEN(SA$)
1010 CALL HCHAR(15,6+I,ASC(S
EG$(SA$,I,1)))
1020 NEXT I
1030 FOR I=1 TO LEN(SB$)
1040 CALL HCHAR(17,6+I,ASC(S
EG$(SB$,I,1)))
1050 NEXT I
1060 CALL KEY(0,K,S)
1070 IF S=0 THEN 1060
1080 IF K=49 THEN 1120
1090 IF K<>50 THEN 1060
1100 SP=2
1110 GOTO 1130
1120 SP=1
1130 CALL CLEAR
1140 CALL SCREEN(2)
1150 FOR I=3 TO 7
1160 CALL COLOR(I,15,1)
1170 NEXT I
1180 CALL COLOR(2,15,8)
1190 CALL COLOR(16,15,8)
1200 CALL COLOR(14,13,8)
1210 CALL COLOR(15,14,4)
1220 CALL COLOR(13,7,8)
1230 CALL COLOR(12,2,8)
1240 CALL COLOR(11,7,4)
1250 S$="SCORE"
1260 FOR I=1 TO LEN(S$)
1270 CALL HCHAR(2,3+I,ASC(SE
G$(S$,I,1)))
1280 NEXT I
1290 M$="MEN"
1300 FOR I=1 TO LEN(M$)
1310 CALL HCHAR(2,15+I,ASC(S
EG$(M$,I,1)))
1320 NEXT I
1330 P=0
1340 MEN=5
1350 BZ=0
1360 GOSUB 3920
1370 GOSUB 4000
1380 IF SP=1 THEN 1410
1390 CALL COLOR(12,2,2)
1400 CALL HCHAR(2,23,121,7)
1410 M=0
1420 Y=0
1430 U=0
1440 L=0
1450 T=0
1460 PU=0
1470 R=0
1480 BL=0
1490 CALL COLOR(9,8,8)
1500 CALL HCHAR(4,1,97,608)
1510 CALL COLOR(10,4,4)
1520 CALL HCHAR(20,1,104,160
)
1530 FOR I=1 TO 100
1540 NEXT I
1550 RANDOMIZE
1560 REM WIND J/N
1570 IF SP=1 THEN 1810
1580 WM=INT(RND*3)+1
1590 IF WM=2 THEN 1630
1600 IF WM=3 THEN 1650
1610 WM$="WIND  O"
1620 GOTO 1660
1630 WM$="WIND <<"
1640 GOTO 1660
1650 WM$="WIND >>"
1660 FOR I=1 TO LEN(WM$)
1670 CALL HCHAR(2,22+I,ASC(S
EG$(WM$,I,1)))
1680 IF WM=1 THEN 1730
1690 IF WM=3 THEN 1720
1700 CALL SOUND(100,280,0)
1710 GOTO 1730
1720 CALL SOUND(100,600,0)
1730 NEXT I
1740 REM POSITION LANDEMARKE

1750 IF WM=1 THEN 1810
1760 IF WM=2 THEN 1790
1770 A=INT(RND*22)+8
1780 GOTO 1820
1790 A=INT(RND*23)+3
1800 GOTO 1820
1810 A=INT(RND*24)+4
1820 CALL HCHAR(24,A,113)
1830 CALL HCHAR(24,A-1,112)
1840 CALL HCHAR(24,A+1,115)
1850 REM POSITION BAEUME
1860 FOR I=1 TO 6
1870 B=18
1880 C=INT(RND*29)+2
1890 CALL GCHAR(B,C,Z)
1900 IF Z<>97 THEN 1870
1910 CALL GCHAR(B,C-1,Z)
1920 IF Z<>97 THEN 1870
1930 CALL GCHAR(B,C+1,Z)
```

```
1940 IF Z<>97 THEN 1870
1950 F=24
1960 CALL GCHAR(F,C,Q)
1970 IF Q<>104 THEN 1870
1980 CALL GCHAR(F,C-1,Q)
1990 IF Q<>104 THEN 1870
2000 CALL GCHAR(F,C+1,Q)
2010 IF Q<>104 THEN 1870
2020 CALL HCHAR(B,C,138)
2030 CALL HCHAR(B+1,C,138)
2040 CALL HCHAR(B+2,C,144)
2050 CALL HCHAR(B+1,C-1,142)
2060 CALL HCHAR(B+1,C+1,143)
2070 CALL HCHAR(B,C-1,140)
2080 CALL HCHAR(B,C+1,141)
2090 CALL HCHAR(B-1,C-1,137)
2100 CALL HCHAR(B-1,C,138)
2110 CALL HCHAR(B-1,C+1,139)
2120 CALL HCHAR(B-2,C,136)
2130 NEXT I
2140 REM WOLKEN
2150 FOR I=1 TO 9
2160 D=5
2170 E=INT(RND*28)+3
2180 CALL GCHAR(D,E,Z)
2190 IF Z<>97 THEN 2160
2200 CALL HCHAR(D,E,40)
2210 CALL HCHAR(D+1,E,42)
2220 CALL HCHAR(D,E+1,41)
2230 CALL HCHAR(D+1,E+1,43)
2240 NEXT I
2250 F=6
2260 G=INT(RND*21)+6
2270 CALL GCHAR(F,G,Z)
2280 IF Z<>97 THEN 2250
2290 CALL HCHAR(F,G,152)
2300 CALL HCHAR(F+1,G,156)
2310 CALL HCHAR(F,G+1,153)
2320 CALL HCHAR(F+1,G+1,157)
2330 CALL HCHAR(F,G+2,154)
2340 CALL HCHAR(F+1,G+2,158)
2350 CALL HCHAR(F,G+3,155)
2360 CALL HCHAR(F+1,G+3,159)
2370 REM FLUG
2380 CALL COLOR(12,2,8)
2390 CALL COLOR(13,7,4)
2400 FOR I=2 TO 45
2410 CALL COLOR(13,7,8)
2420 IF I>31 THEN 2440
2430 CALL SOUND(500,-8,0)
2440 IF I>=31 THEN 2660
2450 IF I>28 THEN 2510
2460 CALL GCHAR(7,I+4,W)
2470 CALL GCHAR(7,I+3,Z)
2480 IF W-Z=1 THEN 2590
2490 IF M<1 THEN 2510
2500 IF M<8 THEN 2700
2510 CALL HCHAR(7,I-1,97)
2520 CALL HCHAR(8,I,97)
2530 CALL HCHAR(7,I,130)
2540 CALL HCHAR(7,I+1,131)
2550 CALL HCHAR(8,I+1,133)
2560 CALL HCHAR(7,I+2,132)
2570 CALL HCHAR(8,I+2,134)
2580 GOTO 2710
2590 CALL HCHAR(7,I-1,97)
2600 CALL HCHAR(7,I,97)
2610 CALL HCHAR(8,I,97)
2620 CALL HCHAR(7,I+1,97)
2630 CALL HCHAR(8,I+1,97)
2640 M=M+1
2650 GOTO 2710
2660 CALL HCHAR(7,30,97,3)
2670 CALL HCHAR(8,29,97,4)
2680 Y=2
2690 GOTO 2710
2700 M=M+1
2710 FOR K=1 TO 10
2720 NEXT K
2730 IF Y=2 THEN 2800
2740 IF Y=1 THEN 2810
2750 CALL KEY(0,K,S)
2760 IF S=0 THEN 3400
2770 IF K<>32 THEN 3400
2780 CALL SOUND(50,600,0)
2790 Y=1
2800 IF U=0 THEN 3400
2810 IF WM=3 THEN 2870
2820 L=I-(U+R)
2830 IF SP=1 THEN 2920
2840 IF WM=1 THEN 2920
2850 IF L<3 THEN 2920
2860 GOTO 2890
2870 L=I-(U-R)
2880 IF L>31 THEN 2920
2890 R=R+1
2900 IF R<3 THEN 2930
2910 R=3
2920 BL=BL+1
2930 IF U>15 THEN 3400
2940 IF U=15 THEN 3130
2950 IF U>12 THEN 3050
2960 CALL VCHAR(7+U,L,97)
2970 IF BL>2 THEN 3060
2980 IF WM=3 THEN 3020
2990 CALL VCHAR(7+U,L+1,97)
3000 CALL VCHAR(8+U,L+1,97)
3010 GOTO 3060
3020 CALL VCHAR(7+U,L-1,97)
3030 CALL VCHAR(8+U,L-1,97)
3040 GOTO 3060
3050 CALL VCHAR(7+U,L,104)
3060 CALL VCHAR(8+U,L,122)
3070 CALL VCHAR(9+U,L,120)
3080 IF 9+U+1>19 THEN 3110
3090 CALL GCHAR(9+U+1,L,V)
3100 IF V<>97 THEN 3260
3110 U=U+1
3120 GOTO 3400
3130 CALL VCHAR(22,L,104)
3140 CALL GCHAR(24,L,X)
3150 IF X=104 THEN 3340
```

```
3160 IF X=113 THEN 3190
3170 PU=1
3180 GOTO 3360
3190 PU=3
3200 CALL SOUND(100,250,0)
3210 CALL SOUND(100,280,0)
3220 CALL SOUND(100,320,0)
3230 CALL SOUND(300,430,0)
3240 U=U+1
3250 GOTO 3400
3260 CALL HCHAR(8+U,L,97)
3270 CALL SOUND(100,210,1)
3280 CALL HCHAR(9+U,L,97)
3290 CALL SOUND(100,180,1)
3300 CALL HCHAR(21,L,120)
3310 CALL SOUND(250,150,1)
3320 U=16
3330 GOTO 3380
3340 CALL SOUND(200,180,0)
3350 GOTO 3380
3360 CALL SOUND(100,250,0)
3370 CALL SOUND(250,425,0)
3380 T=1
3390 U=U+1
3400 NEXT I
3410 IF U>0 THEN 3430
3420 T=1
3430 IF PU=0 THEN 3490
3440 IF PU=1 THEN 3470
3450 P=P+3
3460 GOTO 3480
3470 P=P+1
3480 GOSUB 3910
3490 IF T=0 THEN 3520
3500 MEN=MEN-1
3510 GOSUB 4000
3520 IF BZ=1 THEN 3650
3530 IF P<30 THEN 3650
3540 BZ=1
3550 FOR I=1 TO 5
3560 BO$="BONUS SPRINGER"
3570 FOR K=1 TO LEN(BO$)
3580 CALL HCHAR(10,9+K,ASC(S
EG$(BO$,K,1)))
3590 NEXT K
3600 CALL SOUND(-200,600,0,5
00,0,700,0)
3610 CALL HCHAR(10,9,97,15)
3620 NEXT I
3630 MEN=MEN+1
3640 GOSUB 4000
3650 IF MEN=0 THEN 3670
3660 GOTO 1380
3670 CALL CLEAR
3680 CALL SCREEN(4)
3690 FOR I=3 TO 8
3700 CALL COLOR(I,2,4)
3710 NEXT I
3720 E$="SIE HABEN "&STR$(P)
&" PUNKTE"
3730 FOR I=1 TO LEN(E$)
3740 CALL HCHAR(14,4+I,ASC(S
EG$(E$,I,1)))
3750 NEXT I
3760 FOR I=1 TO 300
3770 NEXT I
3780 N$="NOCH EIN SPIEL ? [J
;N]"
3790 FOR I=1 TO LEN(N$)
3800 CALL HCHAR(20,4+I,ASC(S
EG$(N$,I,1)))
3810 NEXT I
3820 CALL KEY(0,K,S)
3830 IF S=0 THEN 3820
3840 IF K=74 THEN 930
3850 IF K<>78 THEN 3820
3860 CALL CLEAR
3870 FOR I=1 TO 5
3880 PRINT "       AUF WIEDER
SEHEN": : : : : : : : : : :
: : : : : : : : : :
3890 NEXT I
3900 END
3910 REM ZAEHLER PUNKTE
3920 P$=STR$(P)
3930 IF LEN(P$)=3 THEN 3960
3940 P$="0"&P$
3950 GOTO 3930
3960 FOR I=1 TO LEN(P$)
3970 CALL HCHAR(2,10+I,ASC(S
EG$(P$,I,1)))
3980 NEXT I
3990 RETURN
4000 REM ZAEHLER MEN
4010 ME$=STR$(MEN)
4020 CALL HCHAR(2,20,ASC(ME$
))
4030 FOR I=1 TO 100
4040 NEXT I
4050 RETURN
4060 CALL CLEAR
4070 PRINT "EIN HUBSCHRAUBER
 MIT 5 FALL-":"SCHIRMSPRINGE
RN FLIEGT UEBER":"EINE LANDS
CHAFT.": :
4080 PRINT "DURCH DRUECKEN D
ER SPACE-":"TASTE WIRD DER S
PRUNG AUS-":"GELOEST.": :
4090 PRINT "TRIFFT DER SPRIN
GER DIE":"LANDEMARKE NICHT W
IRD EIN":"SPRINGER ABGEZOGEN
.": :
4100 PRINT "TRIFFT ER SIE AM
 RAND WIRD":"EBENFALLS EIN S
PRINGER":"ABGEZOGEN;":"SIE E
RHALTEN ABER 1 PUNKT.": :
4110 PRINT "LANDET ER AUF DE
R MITTE DER":"LANDEMARKE ERH
ALTEN SIE":"3 PUNKTE."
4120 PRINT "DIE ANZAHL DER S
PRINGER":"AENDERT SICH NICHT
.":
```

```
4130 PRINT TAB(24);"ENTER"
4140 CALL KEY(3,K,S)
4150 IF S=0 THEN 4140
4160 IF K<>13 THEN 4140
4170 CALL CLEAR
4180 PRINT "WENN SIE 30 PUNK
TE ERREICHEN":"ERHALTEN SIE
EINEN SPRINGER":"ZUSAETZLICH
": : : : : : : : : : :
4190 PRINT TAB(24);"ENTER"
4200 CALL KEY(3,K,S)
4210 IF S=0 THEN 4200
4220 IF K<>13 THEN 4200
4230 CALL CLEAR
4240 PRINT "SIE KOENNEN ZWIS
CHEN ZWEI":"SPIELVARIANTEN W
AEHLEN.": : : :
4250 PRINT "SPIELART 1:": :
4260 PRINT "ABSPRUNG BEI WIN
DSTILLE.": : :
4270 PRINT "SPIELART 2:": :
4280 PRINT "HIERBEI SIND DIE
 WINDVER-":"HAELTNISSE ZU BE
ACHTEN.":"SIE WERDEN VOR DEM
 START DES":"HUBSCHRAUBERS A
NGEZEIGT.": : : :
4290 PRINT "VIEL VERGNUEGEN
BEIM SPIEL": :
4300 PRINT TAB(24);"ENTER"
4310 CALL KEY(3,K,S)
4320 IF S=0 THEN 4310
4330 IF K<>13 THEN 4310
4340 RETURN
```

BÖRSE


```
1Ø REM********************
11 REM*                  *
12 REM*    TI-MUEHLE     *
13 REM*  Copyright  by   *
14 REM* Juergen Bruehne  *
15 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*    11294 Bytes   *
25 REM********************
26 REM
11Ø REM  BEI VERWENDUNG VON
EINEM  SCHWARZWEISSFERNSEHER
 IST ES SINNVOLL IN ZEILE 85
Ø AUS DER 6 EINE 16 ZU MACHE
N!
12Ø REM ALLE REM-ZEILEN KOEN
NEN WEGGELASSEN WERDEN
13Ø CALL SCREEN(2)
14Ø CALL CLEAR
15Ø PRINT "  JUERGEN BRUEHNE
": :"  AM TANNENGARTEN 15":
:"  3548 AROLSEN-MASSENHAUSE
N": : : : :"  STELLT VOR:":"
  -----------": : :
16Ø FOR X=1 TO 15
17Ø CALL COLOR(X,16,2)
18Ø NEXT X
19Ø FOR X=1 TO 5ØØ
2ØØ NEXT X
21Ø CALL SCREEN(14)
22Ø PRINT : : : : : : : : : :
: : : : : : : : : : : : : : :
:"  IN DER GRUNDWERSION": :
: : : : : : : : : : : : : : :
: : : :
23Ø PRINT : : :"* DREIDIMMEN
SIONALE-MUEHLE ":"  --------
----------------": : :" AUF
DEM:": :" TI-99/4A": :" VON
TEXAS INSTRUMENTS": :,"ENTER
 DRUECKEN":
24Ø CALL KEY(Ø,KE,ST)
25Ø IF KE<>13 THEN 24Ø
26Ø PRINT : : :"  BENOETIGEN
 SIE": :"  DIE SPIELREGELN ?
 (J/N)": :
27Ø CALL KEY(Ø,KE,ST)
28Ø IF ST=Ø THEN 27Ø
29Ø IF KE=78 THEN 52Ø
3ØØ CALL CLEAR
31Ø PRINT "........SPIELREGE
LN":"........-----------": :
 :"SINN DIESES SPIELES IST E
S,": :"SO VIELE MUEHLEN WIE
": :
32Ø PRINT "MOEGLICH ZU ERSCH
LIESSEN !": :"(ES GIBT 76 MO
EGLICHKEITEN":"ZU EINER MUEH
```

```
LE ZU KOMMEN)": : : :,"ENTER
 DRUECKEN": : : :
330 CALL KEY(0,KE,ST)
340 IF ST=0 THEN 330
350 PRINT " EINE MUEHLE BEST
EHT AUS": :" 4 SPIELSTEINEN
,DIE IN": :" EINER REIHE ANG
EORDNET ": :" SEIN MUESSEN."
: :
360 CALL KEY(0,KE,ST)
370 IF ST=0 THEN 360
380 PRINT : : : :" ALSO,WENN
 MAN VIER STEINE": :" (DER G
LEICHEN FARBE)": :" MIT EINE
M LINEAL DURCH": :" IHRE MIT
TELPUNKTE": :
390 PRINT " VERBINDEN KANN ,
HAT MAN": :" EINE MUEHLE": :
" AM ENDE DES SPIELES ": :"
WERTET IHR 'TI' DIE MUEHLEN"
: :
400 PRINT "AUS !": :" ICH WU
ENSCHE IHNEN ": :" EIN SPANN
ENDES SPIEL": :,"ENTER DRUEC
KEN"
410 CALL KEY(0,KE,ST)
420 IF KE<>13 THEN 410
430 CALL CLEAR
440 PRINT "    HALT !": : :"
 BALD HAETTE ICH": :" ES VER
GESSEN !": : : :" FALLS SIE
KEINEN": :" MITSPIELER HABEN
 ,": :
450 PRINT " DANN KOENNEN SIE
 AUCH GEGEN": :" MICH SPIELE
N.": :" DAZU GEBEN SIE NUR '
TI' ": :" ANSTELLE DES ZWEIT
EN"
460 PRINT "SPIELERS EIN.": :
 :"VIEL GLUECK IHR": :,"TI-9
9/4A": :"ENTER DRUECKEN": :
470 CALL KEY(0,KE,ST)
480 IF KE<>13 THEN 470
490 FOR X=1 TO 15
500 CALL COLOR(X,2,1)
510 NEXT X
520 CALL CLEAR
530 PRINT "BITTE NAMEN EINGE
BEN !": : :
540 INPUT "SPIELER 1: ":A$
550 INPUT "SPIELER 2: ":B$
560 FOR X=1 TO 32
570 CALL VCHAR(1,X,32,24)
580 NEXT X
590 CALL SCREEN(4)
600 D(4)=T
610 REM DEFINIERUNGEN !
620 CALL CHAR(130,"FFFFFFFF
FFFFFFF")
630 CALL CHAR(140,"187E7EFFF
F7E7E18")
640 CALL CHAR(150,"187E7EFFF
F7E7E18")
650 CALL COLOR(13,2,13)
660 CALL COLOR(14,7,1)
670 CALL COLOR(15,6,1)
680 CALL COLOR(12,13,13)
690 C$="WEITER MIT"
700 D$=" (SPACE)"
710 E$="MOECHTEN SIE"
720 F$="ES NOCHMAL VERSUCHEN
 ?(J/N)"
730 Q=140
740 R=150
750 T=130
760 CALL CHAR(131,"000000181
8000000")
770 CALL CHAR(141,"E02070AE6
A0E090F")
780 CALL HCHAR(1,1,141,768)
790 H$="MUEHLEN.......MUEHLE
N"
800 REM SPIELAUFBAU
810 CALL CLEAR
820 CALL SCREEN(4)
830 PRINT "  ";A$: :"  ";B$:
840 CALL HCHAR(8,1,120,320)
850 FOR X=1 TO 16
860 READ S
870 READ U
880 CALL VCHAR(18,3+ZA,X+64)
890 CALL VCHAR(S,3+ZA,131,U)
900 ZA=ZA+2
910 IF ZA=8 THEN 940
920 IF ZA=15 THEN 940
930 IF ZA<>22 THEN 960
940 RESTORE 4600
950 ZA=ZA-1
960 NEXT X
970 FOR X=1 TO 16
980 READ S
990 READ Z
1000 CALL VCHAR(S,Z,T,4)
1010 NEXT X
1020 REM ABFRAGE DES BESTIM-
         MUNGSORTES (WIRD
1030 REM MIT DER TASTATUR
1040 REM ABGEFRAGT)
1050 ZA=0
1060 CALL HCHAR(21,3,Q)
1070 CALL HCHAR(23,3,R)
1080 CALL HCHAR(23,24,120)
1090 CALL HCHAR(23,26,120)
1100 CALL HCHAR(21,24,94)
1110 CALL HCHAR(21,26,63)
1120 Y=Q
1130 IF Y<>R THEN 1150
1140 IF B$="TI" THEN 4210
1150 CALL KEY(0,KE,ST)
1160 IF ST=0 THEN 1130
1170 IF KE<65 THEN 1130
1180 IF KE>80 THEN 1130
```

```
1190 ON KE-64 GOTO 1200,1300
,1410,1500,1590,1680,1770,18
60,1950,2040,2130,2220,2310,
2400,2490,2580
1200 CALL GCHAR(13,3,A(4))
1210 IF A(4)<>T THEN 1130
1220 REM SETZEN DER SPIEL-
        STEINE
1230 FOR X=1 TO 4
1240 CALL GCHAR(17-X,3,A(X))
1250 IF A(X)<>T THEN 1280
1260 CALL HCHAR(17-X,3,Y)
1270 GOTO 2670
1280 NEXT X
1290 GOTO 2670
1300 CALL GCHAR(11,5,B(4))
1310 IF B(4)<>T THEN 1130
1320 FOR X=1 TO 4
1330 CALL GCHAR(15-X,5,B(X))
1340 IF B(X)<>T THEN 1370
1350 CALL HCHAR(15-X,5,Y)
1360 GOTO 2670
1370 NEXT X
1380 CALL GCHAR(9,7,C(4))
1390 IF C(4)<>T THEN 1130
1400 GOTO 2670
1410 CALL GCHAR(9,7,C(4))
1420 IF C(4)<>T THEN 1130
1430 FOR X=1 TO 4
1440 CALL GCHAR(13-X,7,C(X))
1450 IF C(X)<>T THEN 1480
1460 CALL HCHAR(13-X,7,Y)
1470 GOTO 2670
1480 NEXT X
1490 GOTO 2670
1500 CALL GCHAR(7,9,D(4))
1510 IF D(4)<>T THEN 1130
1520 FOR X=1 TO 4
1530 CALL GCHAR(11-X,9,D(X))
1540 IF D(X)<>T THEN 1570
1550 CALL HCHAR(11-X,9,Y)
1560 GOTO 2670
1570 NEXT X
1580 GOTO 2670
1590 CALL GCHAR(13,10,E(4))
1600 IF E(4)<>T THEN 1130
1610 FOR X=1 TO 4
1620 CALL GCHAR(17-X,10,E(X)
)
1630 IF E(X)<>T THEN 1660
1640 CALL HCHAR(17-X,10,Y)
1650 GOTO 2670
1660 NEXT X
1670 GOTO 2670
1680 CALL GCHAR(11,12,F(4))
1690 IF F(4)<>T THEN 1130
1700 FOR X=1 TO 4
1710 CALL GCHAR(15-X,12,F(X)
)
1720 IF F(X)<>T THEN 1750
1730 CALL HCHAR(15-X,12,Y)
1740 GOTO 2670
1750 NEXT X
1760 GOTO 2670
1770 CALL GCHAR(9,14,G(4))
1780 IF G(4)<>T THEN 1130
1790 FOR X=1 TO 4
1800 CALL GCHAR(13-X,14,G(X)
)
1810 IF G(X)<>T THEN 1840
1820 CALL HCHAR(13-X,14,Y)
1830 GOTO 2670
1840 NEXT X
1850 GOTO 2670
1860 CALL GCHAR(7,16,H(4))
1870 IF H(4)<>T THEN 1130
1880 FOR X=1 TO 4
1890 CALL GCHAR(11-X,16,H(X)
)
1900 IF H(X)<>T THEN 1930
1910 CALL HCHAR(11-X,16,Y)
1920 GOTO 2670
1930 NEXT X
1940 GOTO 2670
1950 CALL GCHAR(13,17,I(4))
1960 IF I(4)<>T THEN 1130
1970 FOR X=1 TO 4
1980 CALL GCHAR(17-X,17,I(X)
)
1990 IF I(X)<>T THEN 2020
2000 CALL HCHAR(17-X,17,Y)
2010 GOTO 2670
2020 NEXT X
2030 GOTO 2670
2040 CALL GCHAR(11,19,J(4))
2050 IF J(4)<>T THEN 1130
2060 FOR X=1 TO 4
2070 CALL GCHAR(15-X,19,J(X)
)
2080 IF J(X)<>T THEN 2110
2090 CALL HCHAR(15-X,19,Y)
2100 GOTO 2670
2110 NEXT X
2120 GOTO 2670
2130 CALL GCHAR(9,21,K(4))
2140 IF K(4)<>T THEN 1130
2150 FOR X=1 TO 4
2160 CALL GCHAR(13-X,21,K(X)
)
2170 IF K(X)<>T THEN 2200
2180 CALL HCHAR(13-X,21,Y)
2190 GOTO 2670
2200 NEXT X
2210 GOTO 2670
2220 CALL GCHAR(7,23,L(4))
2230 IF L(4)<>T THEN 1130
2240 FOR X=1 TO 4
2250 CALL GCHAR(11-X,23,L(X)
)
2260 IF L(X)<>T THEN 2290
```

```
2270 CALL HCHAR(11-X,23,Y)
2280 GOTO 2670
2290 NEXT X
2300 GOTO 2670
2310 CALL GCHAR(13,24,M(4))
2320 IF M(4)<>T THEN 1130
2330 FOR X=1 TO 4
2340 CALL GCHAR(17-X,24,M(X)
)
2350 IF M(X)<>T THEN 2380
2360 CALL HCHAR(17-X,24,Y)
2370 GOTO 2670
2380 NEXT X
2390 GOTO 2670
2400 CALL GCHAR(11,26,N(4))
2410 IF N(4)<>T THEN 1130
2420 FOR X=1 TO 4
2430 CALL GCHAR(15-X,26,N(X)
)
2440 IF N(X)<>T THEN 2470
2450 CALL HCHAR(15-X,26,Y)
2460 GOTO 2670
2470 NEXT X
2480 GOTO 2670
2490 CALL GCHAR(9,28,O(4))
2500 IF O(4)<>T THEN 1130
2510 FOR X=1 TO 4
2520 CALL GCHAR(13-X,28,O(X)
)
2530 IF O(X)<>T THEN 2560
2540 CALL HCHAR(13-X,28,Y)
2550 GOTO 2670
2560 NEXT X
2570 GOTO 2670
2580 CALL GCHAR(7,30,P(4))
2590 IF P(4)<>T THEN 1130
2600 FOR X=1 TO 4
2610 CALL GCHAR(11-X,30,P(X)
)
2620 IF P(X)<>T THEN 2650
2630 CALL HCHAR(11-X,30,Y)
2640 GOTO 2670
2650 NEXT X
2660 GOTO 2670
2670 ZA=ZA+1
2680 CALL SOUND(-8,560,1)
2690 IF ZA=64 THEN 2780
2700 IF Y<>Q THEN 1080
2710 Y=R
2720 CALL HCHAR(21,24,120)
2730 CALL HCHAR(21,26,120)
2740 CALL HCHAR(23,24,94)
2750 CALL HCHAR(23,26,63)
2760 IF B$="TI" THEN 4210
2770 GOTO 1130
2780 CALL HCHAR(23,24,130,3)
2790 GOSUB 3130
2800 Z=21
2810 S=20
2820 G$=C$
2830 GOSUB 3080
2840 Z=23
2850 G$=D$
2860 GOSUB 3080
2870 CALL KEY(0,KE,ST)
2880 IF KE<>32 THEN 2870
2890 CALL HCHAR(1,1,141,768)
2900 Z=13
2910 S=3
2920 G$=E$
2930 GOSUB 3080
2940 Z=15
2950 G$=F$
2960 GOSUB 3080
2970 CALL KEY(0,KE,ST)
2980 IF ST=0 THEN 2970
2990 IF KE<>74 THEN 3020
3000 RESTORE
3010 GOTO 520
3020 CALL CLEAR
3030 PRINT ".....AUF WIEDERS
EHEN!": : : : : : : :
3040 FOR X=1 TO 200
3050 NEXT X
3060 END
3070 REM   SCHREIBEN DER
           STRINGS
3080 FOR X=1 TO LEN(G$)
3090 CALL HCHAR(Z,S+X-1,ASC(
SEG$(G$,X,1)))
3100 NEXT X
3110 RETURN
3120 REM *MUEHLENAUSWERTUNG*
3130 CALL GCHAR(13,3,A(4))
3140 CALL GCHAR(11,5,B(4))
3150 CALL GCHAR(9,7,C(4))
3160 CALL GCHAR(7,9,D(4))
3170 CALL GCHAR(13,10,E(4))
3180 CALL GCHAR(11,12,F(4))
3190 CALL GCHAR(9,14,G(4))
3200 CALL GCHAR(7,16,H(4))
3210 CALL GCHAR(13,17,I(4))
3220 CALL GCHAR(11,19,J(4))
3230 CALL GCHAR(9,21,K(4))
3240 CALL GCHAR(7,23,L(4))
3250 CALL GCHAR(13,24,M(4))
3260 CALL GCHAR(11,26,N(4))
3270 CALL GCHAR(9,28,O(4))
3280 CALL GCHAR(7,30,P(4))
3290 SW=560
3300 ZZ=0
3310 FOR X=1 TO 6
3320 READ AA,BB,CC,DD
3330 IF A(AA)+B(BB)+C(CC)+D(
DD)<>SW THEN 3350
3340 ZZ=ZZ+1
3350 IF E(AA)+F(BB)+G(CC)+H(
DD)<>SW THEN 3370
3360 ZZ=ZZ+1
3370 IF I(AA)+J(BB)+K(CC)+L(
```

```
DD)<>SW THEN 3390
3380 ZZ=ZZ+1
3390 IF M(AA)+N(BB)+O(CC)+P(
DD)<>SW THEN 3410
3400 ZZ=ZZ+1
3410 IF A(AA)+E(BB)+I(CC)+M(
DD)<>SW THEN 3430
3420 ZZ=ZZ+1
3430 IF B(AA)+F(BB)+J(CC)+N(
DD)<>SW THEN 3450
3440 ZZ=ZZ+1
3450 IF C(AA)+G(BB)+K(CC)+O(
DD)<>SW THEN 3470
3460 ZZ=ZZ+1
3470 IF D(AA)+H(BB)+L(CC)+P(
DD)<>SW THEN 3490
3480 ZZ=ZZ+1
3490 IF A(AA)+F(BB)+K(CC)+P(
DD)<>SW THEN 3510
3500 ZZ=ZZ+1
3510 IF D(AA)+G(BB)+J(CC)+M(
DD)<>SW THEN 3530
3520 ZZ=ZZ+1
3530 NEXT X
3540 IF A(1)+A(2)+A(3)+A(4)<
>SW THEN 3560
3550 ZZ=ZZ+1
3560 IF B(1)+B(2)+B(3)+B(4)<
>SW THEN 3580
3570 ZZ=ZZ+1
3580 IF C(1)+C(2)+C(3)+C(4)<
>SW THEN 3600
3590 ZZ=ZZ+1
3600 IF D(1)+D(2)+D(3)+D(4)<
>SW THEN 3620
3610 ZZ=ZZ+1
3620 IF E(1)+E(2)+E(3)+E(4)<
>SW THEN 3640
3630 ZZ=ZZ+1
3640 IF F(1)+F(2)+F(3)+F(4)<
>SW THEN 3660
3650 ZZ=ZZ+1
3660 IF G(1)+G(2)+G(3)+G(4)<
>SW THEN 3680
3670 ZZ=ZZ+1
3680 IF H(1)+H(2)+H(3)+H(4)<
>SW THEN 3700
3690 ZZ=ZZ+1
3700 IF J(1)+J(2)+J(3)+J(4)<
>SW THEN 3720
3710 ZZ=ZZ+1
3720 IF K(1)+K(2)+K(3)+K(4)<
>SW THEN 3740
3730 ZZ=ZZ+1
3740 IF L(1)+L(2)+L(3)+L(4)<
>SW THEN 3760
3750 ZZ=ZZ+1
3760 IF M(1)+M(2)+M(3)+M(4)<
>SW THEN 3780
3770 ZZ=ZZ+1
3780 IF N(1)+N(2)+N(3)+N(4)<
>SW THEN 3800
3790 ZZ=ZZ+1
3800 IF O(1)+O(2)+O(3)+O(4)<
>SW THEN 3820
3810 ZZ=ZZ+1
3820 IF P(1)+P(2)+P(3)+P(4)<
>SW THEN 3840
3830 ZZ=ZZ+1
3840 ZV=ZV+1
3850 ON ZV GOTO 3860,3910
3860 ZD=ZZ
3870 ZZ=0
3880 SW=600
3890 RESTORE 4640
3900 GOTO 3310
3910 G$=H$
3920 Z=2
3930 S=10
3940 GOSUB 3080
3950 CALL HCHAR(2,5,140)
3960 CALL HCHAR(2,19,150)
3970 IF ZD<20 THEN 4000
3980 CALL HCHAR(2,7,50)
3990 ZD=ZD-20
4000 IF ZD<10 THEN 4030
4010 CALL HCHAR(2,7,49)
4020 ZD=ZD-10
4030 ZD=ZD+48
4040 CALL HCHAR(2,8,ZD)
4050 IF ZZ<20 THEN 4080
4060 CALL HCHAR(2,21,50)
4070 ZZ=ZZ-20
4080 IF ZZ<10 THEN 4110
4090 CALL HCHAR(2,21,49)
4100 ZZ=ZZ-10
4110 ZZ=ZZ+48
4120 CALL HCHAR(2,22,ZZ)
4130 ZZ=0
4140 ZD=0
4150 ZA=0
4160 ZV=0
4170 RESTORE 4640
4180 RETURN
4190 REM BERECHNUNG FUER DEN
'TI-99/4A' ZUR SETZUNG DER
 SPIELSTEINE
4210 IF A(1)+A(2)<>270 THEN
4230
4220 GOTO 1200
4230 IF B(1)+B(2)<>270 THEN
4250
4240 GOTO 1300
4250 IF C(1)+C(2)<>270 THEN
4270
4260 GOTO 1410
4270 IF D(1)+D(2)<>270 THEN
4290
4280 GOTO 1500
4290 IF E(1)+E(2)<>270 THEN
4310
```

```
4300 GOTO 1590
4310 IF F(1)+F(2)<>270 THEN
4330
4320 GOTO 1680
4330 IF G(1)+G(2)<>270 THEN
4350
4340 GOTO 1770
4350 IF H(1)+H(2)<>270 THEN
4370
4360 GOTO 1860
4370 IF I(1)+I(2)<>270 THEN
4390
4380 GOTO 1950
4390 IF J(1)+J(2)<>270 THEN
4410
4400 GOTO 2040
4410 IF K(1)+K(2)<>270 THEN
4430
4420 GOTO 2130
4430 IF L(1)+L(2)<>270 THEN
4450
4440 GOTO 2220
4450 IF M(1)+M(2)<>270 THEN
4470
4460 GOTO 2310
4470 IF N(1)+N(2)<>270 THEN
4490
4480 GOTO 2400
4490 IF O(1)+O(2)<>270 THEN
4510
4500 GOTO 2490
4510 IF P(1)+P(2)<>270 THEN
4530
4520 GOTO 2580
4530 RANDOMIZE
4540 KK=(RND*20)-2
4550 IF KK<1 THEN 4540
4560 IF KK>16 THEN 4540
4570 ON KK GOTO 1200,1300,14
10,1500,1590,1680,1770,1860,
1950,2040,2130,2220,2310,240
0,2490,2580
4580 REM  DATEN FUER DEN
4590 REM BILDAUFBAU
4600 DATA 17,1,15,3,13,5,11,
7
4610 DATA 13,3,11,5,9,7,7,9,
13,10,11,12,9,14,7,16,13,17,
11,19,9,21,7,23,13,24,11,26,
9,28,7,30
4620 REM DATEN ZUR
4630 REM MUEHLENAUSWERTUNG
4640 DATA 1,1,1,1,2,2,2,2,3,
3,3,3,4,4,4,4,1,2,3,4,4,3,2,
1
4650 REM  ENDE DES LISTINGS
```

```
10 REM********************
11 REM*                  *
12 REM* DREIECK & KREIS- *
13 REM*  BERECHNUNGEN    *
14 REM*  Copyright by    *
15 REM* Juergen Bruehne  *
16 REM*                 *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                 *
23 REM  Speicherbelegung *
24 REM*     4279 Bytes   *
25 REM ******************
26 REM
190 CALL CLEAR
200 PRINT " 1-> DREIECKBEREC
HNUNG (^)": : :" 2-> KREISBE
RECHNUNG": : :" 3-> PROGRAMM
ENDE": : : :
210 CALL KEY(0,K,S)
220 IF S=0 THEN 210
230 CALL CLEAR
240 IF K=49 THEN 280
250 IF K=50 THEN 1190
260 IF K>51 THEN 210
270 END
280 PRINT " A=ANKATHETE": :"
 G=GEGENKATHETE": :" H=HYPOT
HENUSE": :" a=ALPHA": :" b=B
ETA": :" g=GAMMA (/\=90')"
290 PRINT : :".........g....
......................./ \...
......................A...G"
300 PRINT "....../.....\....
................a---H---b"
310 PRINT : :"WEITER MIT 'EN
TER'":"RUECKSPRUNG MIT 'R'"
320 CALL KEY(0,K,S)
330 IF S=0 THEN 320
340 IF K=82 THEN 100
350 CALL CLEAR
360 PRINT : :" BITTE WAEHLEN
 SIE !":" ----------------"
: : : :" [1] HYPOTHENUSE + A
NKATHETE": :" [2] HYPOTHENUS
E + GEGENKA.": :
370 PRINT " [3] HYPOTHENUSE
+ ALPHA": :" [4] HYPOTHENUSE
 + BETA": :" [5] ANKATHETE +
 ALPHA": :" [6] ANKATHETE +
BETA": :
380 PRINT " [7] GEGENKATHETE
 + ALPHA": :" [8] GEGENKATHE
TE + BETA": :" [9] ANKATHETE
 + GEGENKA."
390 CALL KEY(0,K,S)
400 IF S=0 THEN 390
410 V=90
420 IF K<49 THEN 390
430 IF K>57 THEN 390
440 CALL CLEAR
```

```
450 ON K-48 GOTO 460,520,570
,640,710,780,850,920,990
460 INPUT "HYPOTHENUSE ?":HY
470 INPUT "ANKATHETE ?":AK
480 IF AK>=HY THEN 470
490 E=HY^2-AK^2
500 GK=E^(1/2)
510 GOTO 1100
520 INPUT "HYPOTHENUSE ?":HY
530 INPUT "GEGENKATHETE ?":G
K
540 IF GK>=HY THEN 530
550 AK=(HY^2-GK^2)^(1/2)
560 GOTO 1100
570 INPUT "HYPOTHENUSE ?":HY
580 INPUT "ALPHA ?":AL
590 IF AL>=V THEN 580
600 CA=COS(AL*(4*ATN(1))/180
)
610 AK=HY*CA
620 GK=(HY^2-AK^2)^(1/2)
630 GOTO 1100
640 INPUT "HYPOTHENUSE ?":HY
650 INPUT "BETA ?":BE
660 IF BE>=V THEN 650
670 CB=COS(BE*(4*ATN(1))/180
)
680 GK=CB*HY
690 AK=(HY^2-GK^2)^(1/2)
700 GOTO 1100
710 INPUT "ANKATHETE ?":AK
720 INPUT "ALPHA ?":AL
730 IF AL>=V THEN 720
740 CA=COS(AL*(4*ATN(1))/180
)
750 HY=AK/CA
760 GK=(HY^2-AK^2)^(1/2)
770 GOTO 1100
780 INPUT "ANKATHETE ?":AK
790 INPUT "BETA ?":BE
800 IF BE>=V THEN 790
810 SB=SIN(BE*(4*ATN(1))/180
)
820 HY=AK/SB
830 GK=(HY^2-AK^2)^(1/2)
840 GOTO 1100
850 INPUT "GEGENKATHETE ?":G
K
860 INPUT "ALPHA ?":AL
870 IF AL>=V THEN 860
880 SA=SIN(AL*(4*ATN(1))/180
)
890 HY=GK/SA
900 AK=(HY^2-GK^2)^(1/2)
910 GOTO 1100
920 INPUT "GEGENKATHETE ?":G
K
930 INPUT "BETA ?":BE
940 IF BE>=V THEN 930
950 CB=COS(BE*(4*ATN(1))/180
)
960 HY=GK/CB
970 AK=(HY^2-GK^2)^(1/2)
980 GOTO 1100
990 INPUT "ANKATHETE ?":AK
1000 INPUT "GEGENKATHETE ?":
GK
1010 HY=(AK^2+GK^2)^(1/2)
1020 GOTO 1100
1030 PRINT "HYPOTHENUSE",HY:
:"ANKATHETE",AK: :"GEGENKAT
HETE",GK: :"ALPHA",AL: :"BET
A",BE: :"GAMMA",90: :"COS a"
,CA: :
1040 PRINT "SIN a",SA: :"TAN
 a",TA: :"COS b",CB: :"SIN b
",SB: :"TAN B",TB
1050 CALL KEY(0,K,S)
1060 IF S=0 THEN 1050
1070 IF K<>82 THEN 360
1080 CALL CLEAR
1090 GOTO 190
1100 CA=AK/HY
1110 SA=GK/HY
1120 CB=SA
1130 SB=CA
1140 TA=SA/CA
1150 TB=SB/CB
1160 AL=ATN(TA)*(180/(4*ATN(
1)))
1170 BE=ATN(TB)*(180/(4*ATN(
1)))
1180 GOTO 1030
1190 PI=(4*ATN(1))
1200 PRINT : : :"WELCHER WER
T IST GEGEBEN ?": :" [1] DUR
CHMESSER(d)": :" [2] RADIUS(
r)": :" [3] UMFANG(u)": :
1210 PRINT " [4] FLAECHENINH
ALT (A)": :" [5] FORMELN": :
 :"FUER pi=";PI: :" [6] WENN
 FUR pi EINE ANDERE": :
1220 PRINT " ZAHL EINGESETZT
 WERDEN SOLL": :" [7] RUECKS
PRUNG": :
1230 CALL KEY(0,K,S)
1240 IF S=0 THEN 1230
1250 IF K<49 THEN 1230
1260 IF K>55 THEN 1230
1270 CALL CLEAR
1280 PRINT "BITTE ZAHL EINGE
BEN": :"DENKEN SIE DARAN FUE
R EIN": :" ',' EIN '.' ZU SE
TZEN": : : :
1290 ON K-48 GOTO 1320,1370,
1420,1470,1520,1300,100
1300 INPUT "fuer pi=:":PI
1310 GOTO 1200
1320 INPUT "DURCHMESSER (d):
":D
1330 R=D/2
1340 U=D*PI
```

```
1350 A=D^2*PI/4
1360 GOTO 1580
1370 INPUT "RADIUS (r):":R
1380 D=2*R
1390 U=D*PI
1400 A=R^2*PI
1410 GOTO 1580
1420 INPUT "UMFANG (u):":U
1430 D=U/PI
1440 R=D/2
1450 A=R^2*PI
1460 GOTO 1580
1470 INPUT "FLAECHENINHALT (
A):":A
1480 R=(A/PI)^(1/2)
1490 D=R*2
1500 U=D*PI
1510 GOTO 1580
1520 CALL CLEAR
1530 PRINT "........FORMELN
:":"........---------": : :"
d=DURCHMESSER": :"r=RADIUS":
 :"u=UMFANG": :"A=FLAECHENIN
HALT"
1540 PRINT : :" d=2*r","r=d/
2": :" u=d*pi","u=2*r*pi": :
" A=r^2*pi","A=d^2*pi/4": :
:"WEITER MIT 'ENTER'": :
1550 CALL KEY(0,K,S)
1560 IF K<>13 THEN 1550
1570 GOTO 1190
1580 CALL CLEAR
1590 PRINT "DURCHMESSER :":"
  d:",D: :"RADIUS :":"   r:",
R: :"UMFANG :":"  u:",U: :"F
LAECHENINHALT :":"  A:",A: :
"WOLLEN SIE EINEN"
1600 PRINT "KREISABSCHNITT B
ERECHNEN ?": :" (J/N) ?":
1610 CALL KEY(0,K,S)
1620 IF S=0 THEN 1610
1630 IF K<>74 THEN 1200
1640 PRINT "BITTE GEBEN SIE
DEN": :"WINKEL EIN !": : :
1650 INPUT "WINKEL : ":W
1660 IF W>360 THEN 1650
1670 U=U*W/360
1680 A=A*W/360
1690 PRINT : : : :"WINKEL :"
,W: :"FLAECHENINHALT :": :"
 Aw:",A: :"UMFANG :": :"  uw
:",U: : :"WEITER MIT 'ENTER'
": :
1700 CALL KEY(0,K,S)
1710 IF K<>13 THEN 1700
1720 GOTO 1200
```


BÖRSE

Achtung ! Ich drucke kostenlos Eure Listings. Rückporto! Programme an Mcihael Wiese, Postfach 3343, 4950 Minden 2

Verkaufe Programme in TI und XBasic. Liste bei Hans Boyer Jr., Eichenstr. 17, 8075 Vohburg gegen RP.

Verk. TI-99 + Ext. Basic + Rec-Kabel + Joyst. + Module: Buchungsjournal, Munch Man, TI Invaders, Chissh. Trail, Addition, Subtr., Multipl. + Basickurs + viel Softw. auf Kass. + viele Bücher v. Zeitschr. Auch einzeln! VB tel. 09171/1716

Suche TI-User, möglichst mit Assemblerkenntnissen im Raum Salzgitter zum Prg- und Moduletausch
Oliver Moldenhauer, Humboldtallee 19, 3320 SZ-1, Tel.: 05341/54411

---

* Suche Adventuremodul * mit Cassetten!
* Tausche Programme * nur Ex.-Basic!
* Tel.: 02103/47208 *

Module f. TI99/4A ges. Angebote an A. Froberg Klopstockstr. 4, 2850 Bremerhaven. P.S. Zahle per Nachnahme

---

Verkaufe/Tausche TI-org.-Laufw. 250. Inv., Parsec je 30,– Suche Speechy-Sy. bis 80,– Tel.: 07240/5300

Verkaufe 3 Original TI-Revue Programmcassetten aus TI-Spezial Paket 4 Hefte 6+7 DM 20,– 02271/65673

Suche für 99/4A preiswert Minimem, 32K Erweiterung extern, J. Rybarski, 02271/65673

Achtung: Habe vor TI-Club zwecks Progr.-Austausch zu gründen. M. Nagler, Tradtstr. 14, 8492 Fürth, 09973/1517

Verkaufe mein Grafik-Adventure „Osus-Shan 1"; Test in Adv-Magazin: Grafik „Sehr gut" deutschsprachig! In Ti-Bsic! Auch andere Spiele und Anwenderprogramme in Gratis. Info bei: W. Mskens, Kranenburgerstr. 224/4180 Goch 6

TI 99/4A+rec+Kabel+Parsec+TI Invader 350 DM, Ex Basic 150 DM, Speechsyn 120 DM, TI Writer 220 DM, TE II 50 DM alle Preise VHG, Rolf Dönselmann Skagerrakstr. 10c, 2960 Aurich Tel. 04941/4101 App. 278

TI 99/4A + Selbstbauer W-Box / Centronics, 32K, Epromprogrammer, GP100, viel Literatur + Programme. 02721/5395

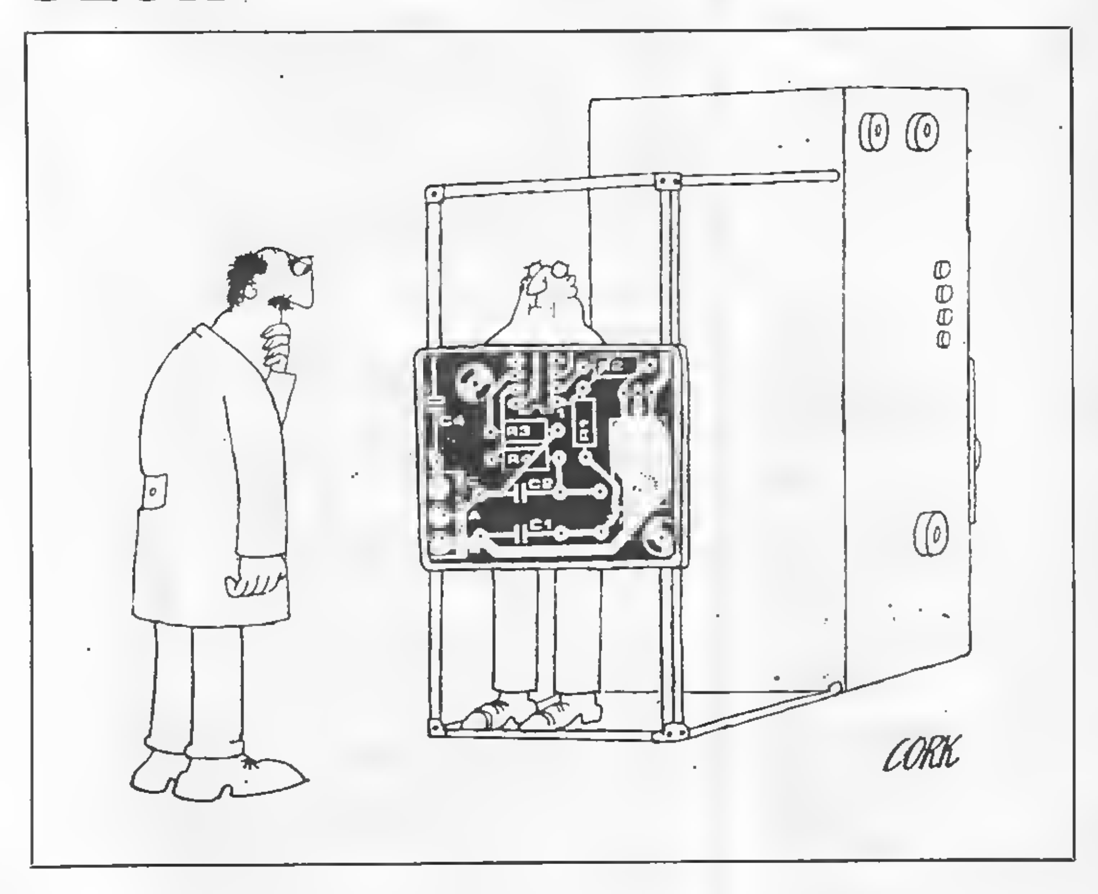
Verk. TI99/4A+Ext. Basic+Rec. Kabel + TI-Joyst. + Handbuch + Data Becker Buch / viele List. u.Progr. Cass. + 8 TI-Revue für DM 400,– Nur komplett. B. Schmitt, Tel. 09721/33843 nach 18 Uhr

---

Suche Amateurfunk-SW jegl. Art: Außerdem: Allg. Literatur zur Assemblerprogrammierung (Bücher/Tips & Tricks)
Hartleib, Hardenbergstr. 16, 3500 Kassel

Exbasic 2 + Painter + Hardcopy, div. Bücher sowie Kass. Kabel günstig !!!
Tel. 06638/1503

BOOGABOO DER MUSIKALISCHE FLOH



Boogaboo ist ein kleiner Floh, und dazu noch ein sehr musikalischer. Deshalb nimmt er alle Gefahren auf sich und sammelt seine verlorengegangenen, verstreuten Noten auf. Da er sich aber in einem vierstöckigen Höhlengewölbe befindet, ist dies nicht allzu leicht. Immerfort drohen fallende Felsbrocken ihn zu erschlagen oder fliegende Notenschlüssel ihn aufzuspießen. Zur Bewegung und zu seiner Hilfe stehen ihm aber Aufzüge und Gleitböden zur Verfügung, auf denen er während des Fahrens völlig sicher ist.

Nach dem Erscheinen des Titelbilds wird gefragt, ob eine Spielerläuterung gewünscht wird. Ist dies nicht der Fall, beginnt das Spiel. Der Spieler steuert Boogaboo mit dem Joystick 2. Durch Drücken des Hebel nach oben oder unten wird der Aufzug benutzt. Er verbindet — bis auf einen Fall im 3. Bild — immer nur 2 Etagen miteinander. Gleichzeitiges Drücken nach rechts oder links und Betätigen des Aktionsknopfes am Joystick bewirkt das Fahren mit dem Gleitboden. Aufgabe ist es, alle verstreuten 18 Noten aufzusammeln, wobei jedoch immer nur drei auf dem Bildschirm erscheinen. Dabei ist sowohl auf die schwindende Zeit als auch auf zahlreiche Hindernisse zu achten. Zum einen muß man aufpassen, daß Boogaboo immer festen Boden unter den Füßen hat, da er ansonsten unweigerlich abstürzt. Zum andern werden ihm fallende Mauerteile und waagerechte Notenschlüssel gefährlich. Lediglich ein aufrechter Notenschlüssel, der jedoch nur zweimal in einem Bild erscheint, bringt Bonuspunkte in Höhe der Hälfte der gegenwärtigen Punktzahl. Fängt Boogaboo diesen Notenschlüssel, so ertönt nach Sammeln aller Noten eine Melodie und die restliche Zeit wird zur Punktzahl addiert. Fängt er ihn nicht, gibt es keinen Bonus und die Melodie kann aufgrund des fehlenden Notenschlüsses nicht angestimmt werden.

Sind alle 18 Noten gesammelt, erscheint ein schwierigeres Bild und Boogaboo beginnt seine Sammelaktion von Neuem. Das Spiel ist beendet, wenn alle 4 Leben unachtsam verspielt worden sind.

Boogaboo ist auf dem TI-99/4A mit EXTENDED Basic programmiert. Ein Großteil der verwendeten Routinen wurde in Unterprogrammen (Subprogramms) erstellt. Dadurch wird neben der Übersichtlichkeit in einigen Fällen auch die Ausführungsgeschwindigkeit erhöht.

Bei der Programmerstellung habe ich bewußt auf effektvolle Graphikhascherei verzichtet. So stellt das vorliegende Programm in Geschwindigkeit, Graphik, Sound und Abwechslung einen — wie ich meine — den Fähigkeiten des TI entsprechenden optimalen Kompromiß dar. In dieser Weise werden denn auch nach Initialisierung ca. 13 600 Bytes belegt, so daß im Interesse eines reibungslosen Programmablaufes davon abzusehen ist, eventuelle größere Veränderungen vorzunehmen. Dementsprechend waren bei mir während des Endstadiums der Entwicklung Programmverluste und Systemabstürze an der Tagesordnung.

Wer dennoch Boogaboo eine größere Chance geben bzw. seine Geschwindigkeit erhöhen will, kann dies durch Ändern der Zeile 600 bewirken. Hier muß nur in den

Bitte lesen Sie weiter auf Seite 123

```
10 !  **********************
11 !  *                    *
12 !  *      BOOGABOO      *
13 !  *                    *
14 !  *    Copyright by    *
15 !  *   Peter Krawinkel  *
16 !  *                    *
19 !  * Benoetigte Geraete *
20 !  *  TI99/4A  Konsole  *
21 !  *     Ext. Basic     *
22 !  *    Joystick (1)    *
23 !  *                    *
26 !  *  Speicherbelegung  *
27 !  *    11065 Bytes     *
28 !  *                    *
29 !  **********************
30 !
220 CALL CLEAR :: CALL SCREE
N(5):: CALL MAGNIFY(3):: RAN
DOMIZE
230 FOR I=1 TO 12 :: CALL CO
LOR(I,16,1):: NEXT I :: CALL
 COLOR(10,16,7)
240 OPTION BASE 1 :: DIM P(3
,6,2),ZE(18),SP(18)
250 !
260 ! CHAR-DEFINITIONEN
270 !
280 CALL CHAR(120,"FF7F3F3F1
20A080D327430121008070OFFFEF
CF8489360920C08682810102OC")
290 CALL CHAR(124,"0040C0201
008040203050F057C23103800040
608102040808040E0407C881038"
)
300 CALL CHAR(128,"000000001
028C44203050F053C43808000000
0001028468480 40E04078840202"
)
310 CALL CHAR(132,"1C08113E0
2070201010204081060200 01C08C
43EA0F0A0C04020100804030 2")
320 CALL CHAR(36,"0102020101
01010709111090701010 6C02020
20408000C020101020C")
330 CALL CHAR(136,"000000060
990907F10100906"&RPT$("0",18
)&"8086FAA2920E")
340 CALL CHAR(140,RPT$("0",1
6)&"FFFF7F3F1F0F"&RPT$("0",2
0)&"FFFFFEFCF8F")
350 CALL CHAR(104,"FF202020F
F020202",112,"FF7F3F3F",113,
"FFFEFCF8",114,"")
360 CALL CHAR(115,"070404040
41C3C18",116,"1F11111111177FF
66",117,"07040404041C2418",1
18,"1F11111111779966",119,"C
34224183C5A42")
370 GOTO 620
380 PU,R,R1=0 :: MA=3
390 !
400 ! RUNDENBEGINN
410 !
420 R=R+1 :: R1=R1+1 :: IF R
>3 THEN R=1
430 AN,NS,A,B,BN,BC=0 :: TI=
250
440 CALL LIFT(R,P(,,),AW())
: CALL NOTVER(ZE(),SP(),R)::
 GOSUB 1140
450 CALL BILD(R):: CALL SPRI
TE(#1,124,16,129,57):: CALL
COLOR(#2,2,#3,2,#4,2,#5,2,#6
,11,#7,11)
460 !
470 ! HAUPTPROGRAMM
480 !
490 CALL TREFF(A,B,BN):: CAL
L COINC(#1,#10,10,CU):: IF C
U THEN 820 ELSE GOSUB 570
500 CALL FNS(A,B,R1,SX,BN,BC
):: CALL COINC(#1,#10,10,CU)
:: IF CU THEN 820 ELSE GOSUB
 570
510 CALL TREFF(A,B,BN):: CAL
L COINC(#1,#10,10,CU):: IF C
U THEN 820 ELSE GOSUB 570
520 CALL FALL(A,B,R1,SY):: C
ALL COINC(#1,#10,10,CU):: IF
 CU THEN 820 ELSE GOSUB 570
530 TI=TI-1 :: IF TI<=0 THEN
 820 ELSE GOTO 490
540 !
550 ! BEWEGUNG
560 !
570 CALL KEY(2,K,S):: CALL J
OYST(2,X,Y):: IF Y<>0 OR S*X
 THEN CALL MOV(R,P(,,),AW(),
Y,X)
580 CALL POSITION(#1,SX,SY):
: CALL GCHAR((SX+7)/8,(SY+11
)/8,P1):: CALL GCHAR((SX+23)
/8,(SY+11)/8,P2)
590 IF P1<>32 THEN GOSUB 121
0 ELSE IF P2=32 THEN 820
600 CALL PATTERN(#1,128):: C
ALL LOCATE(#1,SX,SY+2*X):: C
ALL PATTERN(#1,124):: CALL L
OCATE(#1,SX,SY+2*X)
610 RETURN
620 !
630 ! TITELBILD
640 !
650 CALL CHAR(96,"00000F1F3F
3F3C3C",97,RPT$("3C",8),98,"
0000FFFFFFFF",99,"3C3C3F3F1F
0F",64,"3C4299A1A199423C")
660 CALL CHAR(100,"0000F0F8F
CFC3C3C",101,"3C3CFCFCF8F0",
102,"3C78F0F0F0F0783C",103,"
3C3C3C3C3C3C")
```

```
670 CALL SCREEN(5):: CALL CO
LOR(9,16,1)
680 PRINT "..'bd'bd'bd'b 'bd
'bd'bd'bd"
690 PRINT "..a aa aa aa  a a
a aa aa a"
700 PRINT "..abfa aa aabdaba
abfa aa a"
710 PRINT "..a aa aa aa aa a
a aa aa a"
720 PRINT "..cbecbecbecbeg g
cbecbecbe": : : :"..@ BY P.K
RAWINKEL APRIL'85": : : :
730 PRINT "........hh.......
.hh": : : :"........hh......
..hh": : : : : : :
740 DISPLAY AT(20,3)BEEP:"PR
ESS..." :: DISPLAY AT(22,3)B
EEP:"1 > FOR INSTRUCTIONS" :
: DISPLAY AT(24,3)BEEP:"2 >
TO PLAY "
750 CALL SPRITE(#2,140,2,121
,65,#7,140,11,89,97,#3,140,2
,89,177,#6,140,11,121,145,#1
,124,16,113,153)
760 CALL MUSIC(58)
770 CALL KEY(0,K,S):: IF S=0
 THEN 770 ELSE IF K=49 THEN
1370
780 IF K=50 THEN CALL CLEAR
:: CALL DELSPRITE(ALL):: GOT
O 380 ELSE 770
790 !
800 ! ABSTURZ
810 !
820 CALL DELSPRITE(#10):: IF
 BN=1 THEN 950 ELSE IF TI>0
THEN 840
830 DISPLAY AT(8,4):"DIE ZEI
T WAR BEGRENZT !" :: DISPLAY
 AT(12,4):"WOHL GESCHLAFEN,
ODER ?" :: TI=250
840 FOR I=1 TO 4 :: CALL SOU
ND(-139,110,5,111,5,112,5)::
 CALL PATTERN(#1,128)
850 CALL SOUND(-139,440,5,44
1,5,442,5):: CALL PATTERN(#1
,124):: NEXT I
860 CALL PATTERN(#1,132):: C
ALL MOTION(#1,9,0):: FOR I=6
60 TO 110 STEP -10 :: CALL P
OSITION(#1,SX,SY)
870 CALL SOUND(-139,I,5,I+1,
5,I+2,5):: IF SX>192 THEN 89
0
880 NEXT I
890 CALL DELSPRITE(#1):: FOR
 I=1 TO 4 :: CALL SOUND(169,
110,5,220,5):: NEXT I
900 MA=MA-1 :: IF MA<0 THEN
1290 ELSE A,B=0 :: CALL SCOR
E(PU,HI,MA,TI)
910 CALL HCHAR(8,3,32,26)::
CALL HCHAR(12,3,32,26):: GOT
O 450
920 !
930 ! BONUS
940 !
950 FOR I=1 TO 3 :: FOR J=22
0 TO 330 STEP 10 :: CALL SOU
ND(-139,J,5,J+1,5,J+2,5):: N
EXT J :: NEXT I
960 CALL SPRITE(#11,36,16,17
8,9)
970 FOR I=PU TO PU*1.5 STEP
10*R1 :: DISPLAY AT(22,1)SIZ
E(11):USING "SCORE #####":I
:: CALL SOUND(39,111,5):: NE
XT I
980 BN,A,B=0 :: BC=2 :: PU=P
U*1.5 :: CALL SCORE(PU,HI,MA
,TI):: GOTO 490
990 !
1000 ! NAECHSTE RUNDE
1010 !
1020 CALL DELSPRITE(#10):: C
ALL POSITION(#1,SX,SY)
1030 FOR I=1 TO 6 :: CALL PA
TTERN(#1,128):: CALL LOCATE(
#1,SX-6,SY):: CALL SOUND(-16
9,440,5,444,5)
1040 FOR J=1 TO 35 :: NEXT J
 :: CALL LOCATE(#1,SX,SY)::
CALL PATTERN(#1,124)
1050 CALL SOUND(-169,220,5,2
22,5):: FOR J=1 TO 35 :: NEX
T J :: NEXT I
1060 CALL POSITION(#11,SX,SY
):: IF SX<>178 THEN 1090 ELS
E IF R1/3=INT(R1/3)THEN CALL
 MUSIC(58)ELSE CALL MUSIC(30
)
1070 FOR I=PU TO PU+TI STEP
10 :: DISPLAY AT(22,7)SIZE(5
):USING "#####":I
1080 DISPLAY AT(24,26)BEEP:U
SING "###":PU+TI-I :: NEXT I
 :: PU=PU+TI
1090 TI=0 :: CALL SCORE(PU,H
I,MA,TI):: CALL DELSPRITE(AL
L)
1100 CALL HCHAR(6,1,32,480):
: CALL HCHAR(24,4,32,18):: G
OTO 420
1110 !
1120 ! NOTEN SETZEN
1130 !
1140 NS=NS+3 :: IF NS>=21 TH
EN 1020
1150 FOR I=0 TO 2 :: CALL HC
HAR(ZE(NS-I)*4+1,SP(NS-I),IN
T(RND*4)+115)
```

```
1160 CALL SOUND(-99,220,5,22
1,5,222,5):: NEXT I
1170 TI=TI-3 :: CALL SCORE(P
U,HI,MA,TI):: RETURN
1180 !
1190 ! NOTEN SAMMELN
1200 !
1210 CALL HCHAR((SX+7)/8,(SY
+11)/8,32):: CALL SOUND(-99,
110,5,333,5)
1220 CALL HCHAR(24,NS+AN+1,I
NT(RND*4)+115):: PU=PU+R1*10
 :: TI=TI-1
1230 CALL SCORE(PU,HI,MA,TI)
1240 AN=AN+1 :: IF AN=3 THEN
 AN=0 :: GOSUB 1140
1250 RETURN
1260 !
1270 ! END OF GAME
1280 !
1290 CALL DELSPRITE(ALL):: C
ALL CLEAR :: CALL SCORE(PU,H
I,MA+1,TI):: DISPLAY AT(2,1)
:">>>>>>G A M E  O V E R<<<<
<<"
1300 DISPLAY AT(7,1):"VIER F
LOEHE VERHEIZT !": :"WOHL BL
IND WAS ?"
1310 IF PU>HI THEN CALL SCOR
E(PU,PU,MA+1,TI):: HI=PU ELS
E 1340
1320 DISPLAY AT(12,1):"TROTZ
DEM...................." ::
DISPLAY AT(14,1):"GRATULATIO
N.....HIGHSCORE !!"
1330 CALL MUSIC(58)
1340 DISPLAY AT(17,1):"NEUES
 SPIEL...NEUES GLUECK !": :"
BITTE >J< ODER >N< DRUECKEN!
"
1350 CALL KEY(0,K,S):: IF S=
0 THEN 1350
1360 CALL CLEAR :: IF K=78 O
R K=110 THEN STOP ELSE 380
1370 !
1380 ! SPIELERLAEUTERUNG
1390 !
1400 CALL DELSPRITE(ALL):: C
ALL CLEAR
1410 PRINT "BOOGABOO, DER MU
SIKALISCHE": :"FLOH, WILL SE
INE VERLORENEN": :"NOTEN WIE
DER AUFSAMMELN.": : : :
1420 PRINT "HELFEN SIE IHM D
ABEI, INDEM": :"SIE DIE AUFZ
UEGE (JOYSTICK": :"NACH OBEN
 / UNTEN) ODER": :
1430 PRINT "DIE GLEITBOEDEN
(JOYSTICK": :"RECHTS / LINKS
 + FIRETASTE )": :"BENUTZEN.
": : :
1440 CALL KEY(0,K,S):: IF S=
0 THEN 1440
1450 PRINT "LASSEN SIE SICH
JEDOCH NICHT": :"VON FALLEND
EN MAUERTEILEN": :"ODER LIEG
ENDEN NOTENSCHLUES-": :
1460 PRINT "SELN ERWISCHEN !
": :"NUR AUFRECHTE NOTENSCHL
UES-": :"SEL BRINGEN IHNEN D
IE HAELF-": :
1470 PRINT "TE IHRER PUNKTZA
HL ALS BONUS": :"UND SPAETER
 DIE RESTLICHE": :"ZEIT IN P
UNKTEN.": :
1480 PRINT "DIESER NOTENSCHL
UESSEL ER-": :"SCHEINT IM EI
NER RUNDE NUR": :"2 MAL."
1490 CALL KEY(0,K,S):: IF S=
0 THEN 1490
1500 PRINT : :"OHNE DEN NOTE
NSCHLUESSEL ": :"GIBT'S >KEI
NEN< BONUS, AUCH": :"NICHT A
M RUNDENENDE !": : :
1510 PRINT "EIN TIP !": :"VO
LLKOMMEN SICHER SIND SIE ":
:"NUR, WAEHREND SIE AUFZUG":
 :"ODER GLEITBODEN FAHREN !".
: : : : :
1520 PRINT ">ALPHA-LOCK< LOE
SEN !": :"PRESS ANY KEY TO B
EGIN......"
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530 ELSE CALL CLEAR
:: GOTO 380
1540 !
1550 ! DATEN AUFZUEGE
1560 !
1570 DATA 137,105,32,105,73,
32,73,41,32,73,41,32,81,113,
32,137,153,16
1580 DATA 137,105,32,105,73,
32,73,41,32,73,41,32,57,81,2
4,81,145,64
1590 DATA 137,105,32,105,73,
32,105,41,64,73,41,32,113,16
1,48,97,161,64
1600 !
1610 ! DATEN DISPLAY>1,2,3<
1620 !
1630 DATA "hhhhhhhhh..hhhhh
hhhhhrrhhhh",7
1640 DATA "hhh..hhhhhrrhhhrr
..hhh..hhhh",11
1650 DATA "hhhrrhhhhhhhhhhh
hhh..hhhhhh",15
1660 DATA "hhhhhhhhrr....hhh
hhhrrhhhhhh",19
1670 DATA "hhhhhhhh......hhh
hhhhhhhhhhh",20
1680 DATA 137,177,105,41,73,
97,41,193,137,81,73,137
```

```
1690 DATA "hhh..hhhrr.......
.hhhhrrhhhh",7
1700 DATA "hhhrhhhhhhhhhhhh
h..hh..hhhh",11
1710 DATA "hhhhh...rrhhh..hh
hrrhhhhhhhh",15
1720 DATA "hhhhhhhhhhhhhrrhh
hhhhhhhhhhh",19
1730 DATA "hhhhhhhhhhhhh..hh
hhhhhhhhhhh",20
1740 DATA 137,121,105,161,73
,41,41,193,105,81,41,81
1750 DATA "hhhhhrrhhhhhhhhhhh
hhhhhhhhh..",7
1760 DATA "hhhhh..hhhrr.....
...hh..h",11
1770 DATA "hhhhhhhh..hhrr...
...hhrrhhrr",15
1780 DATA "hhhhhhhhrrhhhhhhh
hhhhhhhhhhh",19
1790 DATA "hhhhhhhh..hhhhhhh
hhhhhhhhhhh",20
1800 DATA 137,81,105,193,105
,225,41,57,105,113,73,97
1810 !
1820 ! DATEN MUSIKNOTEN
1830 !
1840 DATA 2,23,3,10,2,29,1,7
,4,18,3,15,4,29,1,4,2,16,3,1
8,3,4,4,9,1,16,2,23,4,28,3,4
,1,10,1,19
1850 DATA 2,4,4,20,2,19,3,14
,1,22,4,9,1,9,3,5,2,13,4,4,4
,26,1,29,3,6,2,4,1,28,4,24,3
,6,3,27
1860 DATA 1,22,2,27,1,17,3,2
4,4,19,2,10,4,26,3,28,1,4,2,
23,2,4,4,19,3,14,1,23,4,6,2,
27,3,4,3,23
1870 !
1880 ! DATEN MUSIK
1890 !
1900 DATA 5,391,5,523,5,523,
5,587,5,659,5,523,5,659,5,58
7,5,493,5,523,5,523,5,587,5,
659,10,523,5,493
1910 DATA 5,391,5,523,5,523,
5,587,5,659,5,698,5,659,5,58
7,5,523,5,493,5,391,5,440,5,
493,10,523,10,523
1920 DATA 7.5,220,2.5,246,5,
220,5,195,5,220,5,246,5,261,
5,220,7.5,195,2.5,220,5,195,
5,174,10,164,10,195
1930 DATA 7.5,220,2.5,246,5,
220,5,195,5,220,5,246,5,261,
5,220,5,195,5,261,5,246,5,29
3,10,261,5,261
1940 SUB LIFT(R,P(,,),AW())
1950 ON R GOTO 1960,1970,198
0
1960 RESTORE 1570 :: GOTO 19
90
1970 RESTORE 1580 :: GOTO 19
90
1980 RESTORE 1590
1990 FOR I=1 TO 6 :: READ P(
R,I,1),P(R,I,2),AW(I):: NEXT
 I :: SUBEND
2000 SUB BILD(R)
2010 ON R GOTO 2020,2030,204
0
2020 RESTORE 1630 :: GOTO 20
50
2030 RESTORE 1690 :: GOTO 20
50
2040 RESTORE 1750
2050 CALL HCHAR(1,3,104,28):
: DISPLAY AT(2,1):RPT$("xz",
14):: DISPLAY AT(3,1):RPT$("
y(",14)
2060 FOR I=1 TO 5 :: READ A$
,Z :: DISPLAY AT(Z,1):A$ ::
NEXT I
2070 FOR I=2 TO 7 :: READ Z,
S :: CALL SPRITE(#I,140,1,Z,
S):: NEXT I :: SUBEND
2080 SUB MOV(R,P(,,),AW(),Y,
X)
2090 FOR I=2 TO 5 :: CALL CO
INC(#1,#I,8,C(I)):: IF C(I)*
Y THEN 2120
2100 NEXT I
2110 CALL COINC(#1,#6,8,CO1)
:: CALL COINC(#1,#7,8,CO2)::
 IF CO1*X THEN 2150 ELSE IF
CO2*X THEN 2160 ELSE SUBEXIT
2120 CALL POSITION(#I,SX,SY)
:: IF (Y=-4 AND SX=P(R,I-1,1
))OR(Y=4 AND SX=P(R,I-1,2))T
HEN SUBEXIT ELSE 2130
2130 FOR J=SX TO SX-AW(I-1)*
SGN(Y)STEP -Y/2 :: CALL LOCA
TE(#1,J-8,SY,#I,J,SY):: NEXT
 J
2140 CALL HCHAR((SX+15)/8,(S
Y+7)/8,32,2):: CALL HCHAR((S
X+15-SGN(Y)*AW(I-1))/8,(SY+7
)/8,114,2):: SUBEXIT
2150 CALL POSITION(#6,SX,SY)
:: IF (X=4 AND SY=P(R,5,2))O
R(X=-4 AND SY=P(R,5,1))THEN
SUBEXIT ELSE J=5 :: GOTO 217
0
2160 CALL POSITION(#7,SX,SY)
:: IF (X=4 AND SY=P(R,6,2))O
R(X=-4 AND SY=P(R,6,1))THEN
SUBEXIT ELSE J=6
2170 FOR I=SY TO SY+AW(J)*SG
N(X)STEP X/2 :: CALL LOCATE(
#1,SX-8,I,#J+1,SX,I):: NEXT
I
```

```
2180 CALL HCHAR((SX+15)/8,(S
-Y+7)/8,32,2):: CALL HCHAR((S
X+15)/8,(SY+7+SGN(X)*AW(J))/
8,114,2)
2190 SUBEND
2200 SUB FNS(A,B,C,D,E,F)
2210 IF A/2=INT(A/2)OR B=1 T
HEN SUBEXIT ELSE IF F>1 THEN
 PA=136 :: GOTO 2230
2220 IF RND>.26 THEN PA=136
ELSE PA=36 :: E=1 :: F=F+1
2230 H=8*(-1)^(INT(RND*3)+1)
:: IF H>0 THEN G=17 ELSE G=2
39
2240 CALL SPRITE(#10,PA,16,D
-4,G,0,H+SGN(H)*C):: B=1
2250 SUBEND
2260 SUB TREFF(A,B,C)
2270 CALL POSITION(#10,SX,SY
):: IF SY*SX=0 THEN SUBEXIT
ELSE IF SY<18 OR SY>238 OR S
X>145 THEN CALL DELSPRITE(#1
0)ELSE SUBEXIT
2280 A=A+1 :: B,C=0 :: SUBEN
D
2290 SUB FALL(A,B,C,D)
2300 IF A/2<>INT(A/2)OR B=1
THEN SUBEXIT
2310 X=(INT(D+7)/8)*8-7 :: C
ALL SPRITE(#10,120,16,17,X,9
+C,0):: B=1
2320 SUBEND
2330 SUB NOTVER(ZE(),SP(),A)
2340 ON A GOTO 2350,2360,237
0
2350 RESTORE 1840 :: GOTO 23
80
2360 RESTORE 1850 :: GOTO 23
80
2370 RESTORE 1860
2380 FOR I=1 TO 18 :: READ Z
E(I),SP(I):: NEXT I
2390 SUBEND
2400 SUB SCORE(A,B,C,D)
2410 DISPLAY AT(22,1):USING
"SCORE #####  HI ##### CO ##
#":A,B,RPT$("w",C):: DISPLAY
 AT(24,21):USING "TIME ###":
D
2420 SUBEND
2430 SUB MUSIC(A)
2440 RESTORE 1900
2450 FOR I=1 TO A :: READ B,
C :: CALL SOUND(B*38,C,3)::
NEXT I
2460 SUBEND
```

ACHTUNG!
089/1298011 Ab 1.10.85!

BOOGABOO

Fortsetzung von Seite 123

beiden LOCATE-Befehlen der Faktor vor der Statusvariablen ‚X' des Joysticks vergrößert werden. Eventuelle Verbesserungen oder Erweiterungen des Programmes können nach einer Freisetzung von Speicherplatz vorgenommen werden. Dazu kann vielleicht die Spielerläuterung ersatzlos gestrichen und das Titelbild dementsprechend abgeändert werden. Dem Bastler stehen hier alle Wege offen.

Aufbau des Programms BOOGABOO

100–210 Programm Identifizierung
220–240 Dimensionierungen/Farbdefinitionen
250–360 Character-Definitionen
370 Sprung zum Titelbild
380 Variablenbelegung
390–450 Rundenzähler/Variablenbelegung /Bildschirmmaske/Einlesen der Aufzugs- & Noten-Daten
460–530 Hauptprogramm
540–610 Joystick-Abfrage/Bewegung
620–780 Titelbild
790–901 Absturz/ Zeit überschritten
920–980 Bonus
990–1100 Nächste Runde
1110–1170 Noten setzen
1180–1250 Noten sammeln
1260–1360 Spielende
1370–1530 Spielerläuterungen
1540–1600 Daten Aufzüge
1610–1800 Daten Display -1-2-3-
1810–1860 Daten Musik-Noten
1870–1930 Daten Musik
1940–1990 Sub Lift – Einlesen der Aufzugdaten
2000–2070 Sub Bild – Bildschirmmaske
2080–2190 Sub Mov – Aufzüge und Gleitböden bewegen
2200–2250 Sub Fns – Notenschlüssel setzen und bewegen
2260–2280 Sub Treff – Hindernisse löschen
2290–2320 Sub Notver – Notenverteilung einlesen
2400–2420 Sub Score – aktuellen Punktstand drucken
2430–2460 Sub Music – Musik spielen

Variablenliste zum Programm BOOGABOO

Variablen, die ausschließlich in Unterprogrammen benutzt werden, sind mit (S) gekennzeichnet.
A (S) Statusvariable Hindernis
AN Anzahl gesammelter Noten nach dem letzten Setzen
AW (..) (S) Wegstrecke der Aufzüge und Gleitböden
B (S) Hinderniszähler
BC Zähler für Bonuschancen
BN Statusvariable Bonus
Hi Highscore
I,J Schleifenvariablen
MA Restleben
NS Gesamtanzahl gesammelter Noten
PA (S) Gegenwärtiger Mustercode des Notenschlüssels
PU Punkte
P („) (S) Maximale Positionen der Aufzüge und Gleitböden
R Rundenzähler für Bildschirmmaske
R1 Absoluter Rundenzähler
SX/SY Punktzeile / -spalte der Spielerfigur
SX/SY (S) Punktzeile/ -spalte der Aufzüge der Gleitböden
TI Zeit
SP (..) Spalte für Notendruck
ZE(..) Zeile für Notendruck
X,Y Hebelpositionen des Joysticks

Peter Krawinkel

SUPER! DAS SUPER GRAFIK-PROGRAMM



Nach dem Starten des Programmes erscheint das Hauptmenue. Dort können sie zwischen 4 Möglichkeiten wählen. Wählt man Laden oder Speichern, dann wird man nach dem Namen und der Nummer der Datei gefragt. Wählt man Aufruf der Grafik, dann erscheint ein zweites Menue. Dort kann zwischen Erstellen und Definition gewählt werden. Wählt man Definition, so kann man dort mit Hilfe der Tastatur ein Zeichen definieren. Das Umrechnen in den HEX-Code macht der Rechner. Wählt man jedoch Erstellen, so kann man dort eine neue Grafik erstellen oder eine alte Grafik verbessern.

Beschreibung der einzelnen Programmteile:

Hauptprogramm: Hier werden die benötigten Variablen initialisiert.
LOES(chen): Hier wird der Grafikbildschirm gelöscht und gefragt, ob wirklich gelöscht werden soll (LO(esch)FR(age)).
SPRU(ng): Abfrage auf gedrückte Taste und Feststellung, ob eine richtige gedrückt wurde.
LADE: Lädt Grafik von Kassette oder Diskette (FRAG(e)).
SAVE: Speichert Grafik auf Kassette oder Diskette (FRAG(e)).
HAUP(tmenue): Erstellt Menue für das Hauptprogramm.
TITEL: Erstellt Titelbild.
LEER: Leert Bildschirm und initialisiert Farben neu.
GRAF(ik): Fragt, ob man eine Grafik erstellen oder ein Zeichen neu definieren will.
ENDE: Fragt, ob man wirklich beenden will.
ERST(elle): Wenn schon etwas definiert bzw. eine Grafik erstellt wurde, die alte Grafik erstellt (BILD). Beim Drücken einer beliebigen Taste erscheint das Zeichen auf dem Bildschirm unterhalb des Cursors (DRUC(ke)), solange ihr ASCII-Code größer 39 und kleiner 144 ist.

Um den Cursor zu bewegen, müssen die Cursortasten in Verbindung mit FCTN gedrückt werden. Um die Farbe des Bildschirmes zu verändern, muß FCTN REDO drücken. Die Vordergrundfarbe einer Gruppe kann durch FCTN DEL und die Hintergrundfarbe durch FCTN INS erhöht werden. Um diese Veränderungen hervorzurufen, muß ein Zeichen dieser Gruppe unter dem Cursor sein. Mit FCTN BACK kann man dieses Unterprogramm verlassen. Mit FCTN AID erhält man das Menue zu diesem Unterprogramm.
SET (cursor): Zeigt, wo sich der Cursor befindet).
BEST(imme): Bestimmt, von welcher Zeichengruppe die Farbe verändert werden soll.
ERH(öhe): Erhöht oder vermindert einen Wert um 1 bis zu einer maximalen oder minimalen Grenze. Ist eine dieser Grenzen erreicht, so wird der Wert auf die andere Grenze gesetzt.
DEFI(nieren): Hier kann ein Zeichen mit einer 8x8 Matrix definiert werden. Nach der Definition wird die Frage gestellt, ob das Zeichen noch verändert werden soll, oder ob man en neues Zeichen definieren will.
D(efi)MEN(ue): Erstellt das Menue zum Unterprogramm DEFI.
UMRE(chnen): Rechnet die Matrix in den dazugehörigen Hex-Code um.

Bedeutung der wichtigsten Variablen:

ZE$( ): Enthält den Teil des Grafikbildschirms, der mit DISPLAY AT dargestellt werden kann.
CH$ ( ): Enthält die Hex-Codes der neu definierten Zeichen.
SP$ ( ): Enthält die Spalten, die nur mit HCHAR dargetellt werden können.
VO ( ): Enthält die Vordergrundfarben der Zeichengruppen.
HI ( ): Enthält die Hintergrundfarben der Zeichengruppen.
SC: Enthält die Bildschirmfarbe.
DE: Gibt an, ob eine alte Grafik im Speicher ist.
I,J: Zählvariablen
ME$: Gibt die erlaubten Tasten an.

```
10 ! **********************
11 ! *                    *
12 ! *    SUPER-GRAF I    *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *     Uwe Klosa      *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *  Disk-Drive oder   *
23 ! *   Kass.-Rekorder   *
24 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     7591 Bytes     *
28 ! *                    *
29 ! **********************
100 ON WARNING NEXT :: OPTIO
N BASE 1 :: ON BREAK NEXT ::
 CALL TITEL :: RANDOMIZE
110 DIM ZE$(24),CH$(104),SP$
(4),VO(14),HI(14):: DE=-1
120 CALL LOES(ZE$(),CH$(),SP
$(),VO(),HI(),DE,SC)
130 CALL HAUP
140 ME$="SLDEG" :: CALL SPRU
(ME$,K)
150 IF K=83 THEN CALL SAVE(Z
E$(),CH$(),SP$(),VO(),HI(),S
C)
160 IF K=76 THEN CALL LADE(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE)
170 IF K=71 THEN CALL GRAF(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE):: CALL DELSPRITE(#1)
180 IF K=68 THEN 120
190 IF K=69 THEN CALL ENDE
200 IF K=0 THEN 140 ELSE 130
210 SUB LOES(ZE$(),CH$(),SP$
(),VO(),HI(),DE,SC)
220 IF DE>-1 THEN CALL LOFR(
WE):: IF WE=0 THEN SUBEXIT
230 FOR I=1 TO 24 :: ZE$(I)=
RPT$(" ",28):: NEXT I :: FOR
 I=1 TO 4 :: SP$(I)=RPT$(" "
,24):: NEXT I
240 FOR I=1 TO 14 :: VO(I)=2
 :: HI(I)=1 :: NEXT I :: SC=
8 :: FOR I=1 TO 104 :: CH$(I
)="000"&RPT$(" ",16):: NEXT
I :: DE=0
250 SUBEND
260 SUB SPRU(ME$,K)
270 CALL KEY(3,K,S):: IF S=0
 THEN 270
280 FOR I=1 TO LEN(ME$):: IF
 K=ASC(SEG$(ME$,I,1))THEN SU
BEXIT
290 NEXT I :: CALL SOUND(10,
110,15):: K=0
300 SUBEND
310 SUB HAUP
320 CALL LEER :: RESTORE 340
330 FOR I=1 TO 5 :: READ A$
:: DISPLAY AT(2*I+6,5):A$ ::
 NEXT I
340 DATA S> GRAFIK SPEICHERN
,L> GRAFIK LADEN,G> AUFRUF D
ER GRAFIK,D> GRAFIK LOESCHEN
,E> ENDE
350 SUBEND
360 SUB SAVE(ZE$(),CH$(),SP$
(),VO(),HI(),SC)
370 CALL FRAG(NA$,NU):: OPEN
 #NU:NA$,OUTPUT,SEQUENTIAL,I
NTERNAL,FIXED 80
380 FOR I=1 TO 24 STEP 3 ::
PRINT #NU:ZE$(I),ZE$(I+1),ZE
$(I+2):: NEXT I
390 FOR I=1 TO 14 STEP 2 ::
PRINT #NU:VO(I),HI(I),VO(I+1
),HI(I+1):: NEXT I
400 PRINT #NU:SP$(1),SP$(2),
SP$(3),SP$(4),SC
410 FOR I=1 TO 104 STEP 8 ::
 PRINT #NU:CH$(I),CH$(I+1),C
H$(I+2),CH$(I+3),CH$(I+4),CH
$(I+5),CH$(I+6),CH$(I+7):: N
EXT I :: CLOSE #NU
420 SUBEND
430 SUB FRAG(NA$,NU)
440 CALL LEER :: DISPLAY AT(
1,1):"D(ISKETTE ODER K(ASSET
TE ?:" :: ACCEPT AT(1,28)VAL
IDATE("DKdk"):A$
450 IF A$="D" OR A$="d" THEN
 NA$="DSK1" ELSE NA$="CS1"
460 DISPLAY AT(3,1):"DATEINU
MMER ?:" :: ACCEPT AT(3,14)V
ALIDATE(DIGIT):NU :: IF NU<0
 OR NU>255 THEN 460
470 DISPLAY AT(5,1):"DATEINA
ME ?:" :: ACCEPT AT(5,13)VAL
IDATE(UALPHA,DIGIT)SIZE(10):
N$ :: NA$=NA$&"."&N$
480 SUBEND
490 SUB LADE(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
500 CALL FRAG(NA$,NU):: OPEN
 #NU:NA$,INPUT ,SEQUENTIAL,I
NTERNAL,FIXED 80
510 FOR I=1 TO 24 STEP 3 ::
INPUT #NU:ZE$(I),ZE$(I+1),ZE
$(I+2):: NEXT I
520 FOR I=1 TO 14 STEP 2 ::
INPUT #NU:VO(I),HI(I),VO(I+1
),HI(I+1):: NEXT I
530 INPUT #NU:SP$(1),SP$(2),
SP$(3),SP$(4),SC
```

```
540 FOR I=1 TO 104 STEP 8 ::
 INPUT #NU:CH$(I),CH$(I+1),C
H$(I+2),CH$(I+3),CH$(I+4),CH
$(I+5),CH$(I+6),CH$(I+7):: N
EXT I :: CLOSE #NU
550 SUBEND
560 SUB LEER
570 CALL CLEAR :: CALL SCREE
N(5):: FOR I=1 TO 8 :: CALL
COLOR(I,2,16):: NEXT I
580 SUBEND
590 SUB ENDE
600 CALL LEER :: DISPLAY AT(
1,1):"WOLLEN SIE WIRKLICH BE
ENDEN?"
610 CALL SPRU("JN",K):: IF K
=74 THEN END ELSE IF K=78 TH
EN SUBEXIT ELSE 610
620 SUBEND
630 SUB LOFR(WE)
640 CALL LEER :: DISPLAY AT(
1,1):"GRAFIK WIRKLICH LOESCH
EN ?"
650 CALL SPRU("JN",K):: IF K
=74 THEN WE=1 ELSE IF K=78 T
HEN WE=0 ELSE 650
660 SUBEND
670 SUB GMEN
680 CALL LEER :: RESTORE 700
690 FOR I=1 TO 3 :: READ A$
:: DISPLAY AT(2*I+8,5):A$ ::
 NEXT I
700 DATA D> ZEICHENDEFINITIO
N,G> GRAFIK ERSTELLEN,Q> QUI
T
710 SUBEND
720 SUB GRAF(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
730 CALL CHARSET :: CALL GME
N :: CALL SPRU("DGQ",K):: IF
 K=68 THEN CALL DEFI(CH$()):
: DE=1 :: CALL DELSPRITE(#1)
740 IF K=71 THEN CALL ERST(Z
E$(),CH$(),SP$(),VO(),HI(),S
C,DE):: CALL DELSPRITE(#1)::
 DE=1 ELSE IF K=81 THEN SUBE
XIT
750 GOTO 730
760 SUBEND
770 SUB ERST(ZE$(),CH$(),SP$
(),VO(),HI(),SC,DE)
780 X=16 :: Y=12 :: CALL CHA
R(33,"00003C3C3C3C"):: CALL
BILD(ZE$(),CH$(),SP$(),VO(),
HI(),SC,DE)
790 CALL SETC(X,Y,SC):: CALL
 KEY(0,K,S):: IF S=0 THEN 79
0
800 IF K=3 THEN CALL BEST(I,
X,Y):: CALL ERH(VO(I),1,16):
: CALL COLOR(I,VO(I),HI(I))E
LSE IF K=4 THEN CALL BEST(I,
X,Y):: CALL ERH(HI(I),1,16):
: CALL COLOR(I,VO(I),HI(I))
810 IF K=6 THEN CALL ERH(SC,
1,16):: CALL SCREEN(SC)ELSE
IF K=15 THEN SUBEXIT ELSE IF
 K=1 THEN CALL EMEN(ZE$(),CH
$(),VO(),HI(),SC)
820 IF K=8 THEN X=-X :: CALL
 ERH(X,1,32)ELSE IF K=9 THEN
 CALL ERH(X,1,32)
830 IF K=10 THEN CALL ERH(Y,
1,24)ELSE IF K=11 THEN Y=-Y
:: CALL ERH(Y,1,24)
840 IF K>39 AND K<144 THEN C
ALL DRUC(K,X,Y,ZE$(),SP$()):
: GOTO 790 ELSE 790
850 SUBEND
860 SUB ERH(I,MI,MA)
870 IF I>0 THEN D=1 ELSE D=-
1 :: I=ABS(I)
880 I=I+D :: IF I<MI THEN I=
MA ELSE IF I>MA THEN I=MI
890 SUBEND
900 SUB SETC(X,Y,SC)
910 CALL COLOR(#1,1):: X1=X*
8-7 :: Y1=Y*8-7 :: IF SC>2 T
HEN FA=2 ELSE FA=16
920 CALL SPRITE(#1,33,FA,Y1,
X1)
930 SUBEND
940 SUB DRUC(K,X,Y,ZE$(),SP$
())
950 CALL HCHAR(Y,X,K):: IF X
<3 THEN SP$(X)=SEG$(SP$(X),1
,Y-1)&CHR$(K)&SEG$(SP$(X),Y+
1,24-Y)
960 IF X>30 THEN SP$(X-28)=S
EG$(SP$(X-28),1,Y-1)&CHR$(K)
&SEG$(SP$(X-28),Y+1,24-Y)
970 IF X>2 AND X<31 THEN ZE$
(Y)=SEG$(ZE$(Y),1,X-3)&CHR$(
K)&SEG$(ZE$(Y),X-1,30-X)
980 SUBEND
990 SUB BEST(A,X,Y)
1000 CALL GCHAR(Y,X,B):: A=I
NT(B/8)-3
1010 SUBEND
1020 SUB DEFI(CH$())
1030 ZE=33 :: CALL CHAR(33,"
FF8181818181 81FF",34,"FFFFFF
FFFFFFFFFF"):: X=13 :: Y=1
1040 CALL LEER :: FOR I=1 TO
 8 :: FOR J=1 TO 8 :: CALL H
CHAR(I,J+12,ZE):: NEXT J ::
NEXT I :: CALL DMEN
1050 CALL SETC(X,Y,2):: CALL
 SPRU("10ESDXUQ",K):: IF K=4
9 THEN ZE=34 ELSE IF K=48 TH
EN ZE=33 ELSE IF K=81 THEN S
UBEXIT
```

```
1060 IF K=69 THEN Y=-Y :: CA
LL ERH(Y,1,8)ELSE IF K=88 TH
EN CALL ERH(Y,1,8)ELSE IF K=
83 THEN X=-X :: CALL ERH(X,1
3,20)ELSE IF K=68 THEN CALL
ERH(X,13,20)
1070 IF K=85 THEN CALL UMRE(
CH$())ELSE CALL HCHAR(Y,X,ZE
):: GOTO 1050
1080 DISPLAY AT(20,1):"VERAE
NDERUNGEN VORNEHMEN ?" :: CA
LL SPRU("JN",K):: IF K=74 TH
EN CALL DMEN :: GOTO 1050 EL
SE IF K=0 THEN 1080
1090 DISPLAY AT(22,1):"NEUES
 ZEICHEN ?" :: CALL SPRU("JN
",K):: IF K=74 THEN 1030 ELS
E IF K=0 THEN 1090
1100 SUBEND
1110 SUB UMRE(CH$())
1120 DISPLAY AT(18,1):"WELCH
ES ZEICHEN(ASCII) ?" :: ACCE
PT AT(18,26)VALIDATE(DIGIT)S
IZE(3):ZE :: IF ZE<40 OR ZE>
143 THEN 1120
1130 CH$(ZE-39)=RPT$("0",3-L
EN(STR$(ZE)))&STR$(ZE)&""
1140 FOR I=1 TO 8 :: C$="" :
: FOR J=13 TO 16 :: CALL GCH
AR(I,J,G):: C$=C$&CHR$(G)::
NEXT J
1150 CALL WERT(C$,CH$(ZE-39)
):: C$="" :: FOR J=17 TO 20
:: CALL GCHAR(I,J,G):: C$=C$
&CHR$(G):: NEXT J
1160 CALL WERT(C$,CH$(ZE-39)
):: NEXT I :: DISPLAY AT(12,
1):SEG$(CH$(ZE-39),4,16)
1170 SUBEND
1180 SUB WERT(C$,D$)
1190 E=0 :: FOR I=0 TO 3 ::
IF SEG$(C$,4-I,1)=CHR$(34)TH
EN D=2^I ELSE D=0
1200 E=E+D :: NEXT I :: H$="
0123456789ABCDEF"
1210 D$=D$&SEG$(H$,E+1,1)
1220 SUBEND
1230 SUB BILD(ZE$(),CH$(),SP
$(),VO(),HI(),SC,DE)
1240 CALL CLEAR :: IF DE=0 T
HEN 1280
1250 FOR I=1 TO 104 :: V=VAL
(SEG$(CH$(I),1,3)):: IF V>39
 THEN CALL CHAR(V,SEG$(CH$(I
),4,16))
1260 NEXT I :: FOR I=1 TO 24
 :: DISPLAY AT(I,1):ZE$(I)::
 NEXT I
1270 FOR I=1 TO 2 :: FOR J=1
 TO 24 :: CALL HCHAR(J,I,ASC
(SEG$(SP$(I),J,1))):: CALL H
CHAR(J,I+30,ASC(SEG$(SP$(I+2
),J,1))):: NEXT J :: NEXT I
1280 CALL SCREEN(SC):: FOR I
=1 TO 14 :: CALL COLOR(I,VO(
I),HI(I)):: NEXT I
1290 SUBEND
1300 SUB DMEN
1310 RESTORE 1330
1320 FOR I=1 TO 4 :: READ A$
 :: DISPLAY AT(I*2+16,1):A$
:: NEXT I
1330 DATA 0> PUNKT AUS...1>
PUNKT EIN,E> HOCH........X>
RUNTER,S> LINKS.......D> REC
HTS
1340 DATA U> UMRECHNEN...Q>
QUIT
1350 SUBEND
1360 SUB TITEL
1370 CALL CLEAR :: CALL SCRE
EN(2):: FOR I=1 TO 14 :: CAL
L COLOR(I,INT(RND*13+3),2)::
 NEXT I
1380 FOR I=1 TO 32 :: CALL H
CHAR(1,I,42):: CALL HCHAR(24
,I,42):: NEXT I
1390 FOR I=1 TO 24 :: CALL H
CHAR(I,1,42):: CALL HCHAR(I,
32,42):: NEXT I :: DISPLAY A
T(5,9):"TERRA - SOFT" :: DIS
PLAY AT(7,9):"Praesentiert"
1400 DISPLAY AT(10,8):"SUPER
 - GRAF I" :: DISPLAY AT(11,
8):"==============" :: DISPL
AY AT(14,1):"fuer TI99/4A +
EXT + Kas/Dis"
1410 DISPLAY AT(16,1):"Idee
& Ausfuehrung:Uwe Klosa" ::
DISPLAY AT(20,5):"(c) Uwe Kl
osa   1985"
1420 FOR I=1 TO 10 :: FOR J=
1 TO 14 :: CALL COLOR(J,INT(
RND*13+3),2):: NEXT J :: NEX
T I
1430 SUBEND
1440 SUB EMEN(ZE$(),CH$(),VO
(),HI(),SC)
1450 RESTORE 1460 :: CALL CH
ARSET :: CALL SCREEN(16):: F
OR I=1 TO 5 :: READ A$ :: DI
SPLAY AT(I+19,1):A$ :: NEXT
I
1460 DATA FCTN E>HOCH // FCT
N X>RUNTER,FCTN S>LINKS / FC
TN D>RECHTS,FCTN 1>VOR+1 //
FCTN 2>HIN+1
1470 DATA FCTN 8>SCFAR+1 / F
CTN 9>QUIT,FCTN 7>AUFRUF DIE
SES MENUES
1480 CALL KEY(0,K,S):: IF S=
0 THEN 1480
```

```
1490 FOR I=1 TO 56 :: V=VAL(
SEG$(CH$(I),1,3)):: IF V>39
THEN CALL CHAR(V,SEG$(CH$(I)
,4,16))
1500 NEXT I :: FOR I=1 TO 14
 :: CALL COLOR(I,VO(I),HI(I)
)::: NEXT I :: CALL SCREEN(SC
)::: CALL CHAR(33,"00003C3C3C
3C")
1510 FOR I=19 TO 24 :: DISPL
AY AT(I,1):ZE$(I):: NEXT I
1520 SUBEND
```

DER TI ALS SCHREIBMASCHINEN LEHRER

Das Programm Touch Typing Teacher ist ein Schreibmaschinenlehrgang für den TI99/4A.

Es nutzt fast allen graphischen Möglichkeiten des TI99/4A und auch der Ton kommt nicht zu kurz. Wenn man intensiv mit dem Programm arbeitet, kann man nach einiger Zeit die Tastatur des TI fast blind beherrschen, aber auch wenn man sich nicht so intensiv mit dem Programm beschäftigt, ist es nicht abzustreiten, daß man die Tastatur wesentlich besser beherrscht als vorher und sich auch die Schreibgeschwindigkeit stark gesteigert hat. Das Programm stellt also eine wirkliche Alternative zu dem von Texas Instruments vertriebenen Programm Touch Typing Tutor dar.

Der Benutzer kann selbst entscheiden, wann er genug von einer Übung hat, oder, wenn er glaubt, eine Übung gut genug zu beherrschen, kann er sich einfach eine andere Übung heraussuchen.

Aber natürlich kann man mit dem Programm nicht nur üben, man kann z.B. auch auf Zeit schreiben (der Computer mißt die Zeit), oder Texte nachschreiben, die der Computer dann auf ihre Richtigkeit (orthographische) überprüft.

Das Programm besitzt eine benutzerfreundliche Menütechnik, die es jedem Benutzer kinderleicht macht, mit dem Programm umzugehen.

Nach dem Laden des Programms und Eingabe von RUN oder vor dem Laden von RUN „CS1" erscheint nach ein paar Sekunden das erste Titelbild. Nach Drücken einer Taste wird der Bildschirm schwarz, und nach cirka 15 Sekunden erscheint das eigentliche Titelbild, mit der Tastatur des Ti99/4A, auf dem Bildschirm. Wenn Sie nun eine Taste drücken, erscheint das Hauptmenü. Das Hauptmenü umfaßt 6 Wahlmöglichkeiten. Wenn Sie aus irgendeinem Unterprogramm wieder zum Hauptmenü zurückkehren wollen, dann müssen Sie nur bei einer CALL KEY Abfrage FCTN 9 eingeben (drücken).

Wenn Sie nun Punkt 1 drücken, können Sie den Umgang mit der Tastatur etwas üben. Dazu druckt der Computer die Tastatur auf dem Bildschirm aus. Dann wird kurz erklärt, mit welchen Tasten Sie in welches Unterprogramm kommen. Mit FCTN 8 können Sie den Computer dazu veranlassen, daß er die einzelnen Symbole auf die Tastatur druckt, da diese ja nicht vorhanden sind. Sind die Symbole gedruckt, können Sie die Symbole wieder löschen. Dieses Unterprogramm dient nur als Gedankenstütze. Mit FCTN 9 können Sie zum Hauptmenü zurückkehren. Wenn Sie aber nun irgendeine andere Taste drücken, beginnt der Computer mit dem „Kennenlernen der Tastatur". Auf dem Bildschirm ist nun die Tastatur des Ti99/4A zu sehen, allerdings ohne Symbole. Wenn Sie nun irgendeine Taste drücken, erscheint das Symbol, welches auf der Taste zu sehen ist, auf der entsprechenden Taste auf dem Bildschirm und verschwindet kurz danach wieder. Wollen Sie dieses Symbol also länger sehen, müssen Sie nur den Finger länger auf der entsprechenden Taste lassen.

Mit diesem Unterprogramm können Sie lernen, wo die einzelnen Tasten liegen. Wie schon beschrieben, können Sie mit FCTN 9 zurück ins Hauptmenü und mit FCTN 8 die Symbole auf die Tasten drucken lassen. Bei diesem Unterprogramm und bei allen anderen sollten Sie versuchen, nicht auf die Tastatur zu sehen, da sonst der Lernerfolg nicht sehr groß ist.

Folgende Symbole können durch Drücken der entsprechenden Taste nicht auf den Bildschirm gebracht werden (gilt nur für 1 und 2 aus dem Hauptmenü):
alle Kleinbuchstaben (gilt auch für 3 und 4)
die Zeichen Links geschweifte Klammer, Doppelstrich, Rechts geschweifte Klammer, Tilde, Del

Wenn Sie eines dieser Zeichen eingeben, ertönt ein Warnton (nur bei 1 und 2), bei 3 und 4 erscheinen andere Zeichen, da die ASCII-Codes dieser Zeichen mit anderen Zeichen belegt wurden (CALL CHAR).

Wenn Sie 2 drücken, kommen Sie in ein Unterprogramm, welches überprüft, wie gut Sie sich auf der TI-Tastatur auskennen. Auf dem Bildschirm erscheint nun wiederum die TI-Tastatur ohne Symbole. Allerdings erscheint nun immer ein Symbol auf der Bildschirmtastatur, welches Sie auf der TI-Tastatur finden müssen (blind, ohne auf die Tastatur zu sehen). Wenn Sie die richtige Taste gedrückt haben, ertönt eine kurze „Siegesmelodie" und es erscheint ein neues Symbol. Wenn Sie allerdings falsch gedrückt haben, erscheint das falsche Symbol nur für kurze Zeit auf der Bildschirmtastatur und es erklingt eine kleine Melodie, die den falschen Tastendruck akustisch erkennen läßt. Falls Sie eine der nicht erlaubten Tasten gedrückt haben (siehe bei 1 alle Zeichen mit einem ASC II Code über 96), dann ertönt ein kurzer Warnton.

Wenn sie nun einige Zeit mit dem Unterprogramm gearbeitet haben und Sie wollen zum Hauptmenü zurückkehren, müssen Sie ebenfalls

Bitte lesen Sie weiter auf Seite 133

```
10 ! *********************
11 ! *                   *
12 ! *TOUCH TYPING TEACHER*
13 ! *                   *
14 ! *    Copyright by   *
15 ! *   Dirk Junghans   *
16 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                   *
26 ! *  Speicherbelegung  *
27 ! *     10186 Bytes    *
28 ! *                   *
29 ! *********************
30 !
100 OPTION BASE 1 :: GOTO 22
0
110 CALL CHAR :: CALL HCHAR
:: CALL CLEAR :: CALL SCREEN
 :: CALL COLOR :: CALL CHARS
ET :: CALL SOUND :: CALL KEY
120 CALL TASTA :: CALL SONDE
R :: CALL ANZ :: CALL EING :
: CALL ABFEING
130 JN$=""
140 I,S,SE,WA,DJ,DD,I,K,S,X1
,Y1,ZAE,I1,I2,TI
150 DATA ABC
160 !@P-
170 !*bei fragen:          *
180 !*dirk junghans        *
190 !*am fort-biehler 9    *
200 !*6503 mz-kastel       *
210 !**********************
220 CALL CLEAR
230 CALL SCREEN(10)
240 FOR I=2 TO 8 :: CALL COL
OR(I,5,10):: NEXT I
250 CALL COLOR(13,2,10,14,2,
10)
260 CALL CHAR(131,"00"&RPT$(
"01",7),132,"FC8485858484868
2",133,"008040408000000C")
270 CALL CHAR(134,"01FF80C04
060381C",135,"8C9921213D0505
05",136,"12BA8A8ABAA1A1A1")
280 CALL CHAR(137,"0E0301",1
38,"C4E23110180C0703",139,"2
24C90204040A0E0")
290 CALL CLEAR
300 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(3,6):"EIN PROGR
AMM VON" :: DISPLAY AT(5,8):
"DIRK JUNGHANS"
310 DISPLAY AT(8,5):"FUER DE
N TI99/4A VON" :: DISPLAY AT
(16,6):"TEXAS INSTRUMENTS"
320 DISPLAY AT(19,1):"DAS PR
OGRAMM WURDE DER......KLASSE
 10B DER HLS WIESBADENGEWIDM
ET."
330 DISPLAY AT(11,13):CHR$(1
31);CHR$(132);CHR$(133):: DI
SPLAY AT(12,13):CHR$(134);CH
R$(135);CHR$(136)
340 DISPLAY AT(13,13):CHR$(1
37);CHR$(138);CHR$(139)
350 DISPLAY AT(24,3):"PRESS
A KEY TO CONTINUE"
360 CALL KEY(0,K,S):: IF S=0
 THEN 360
370 CALL CLEAR :: CALL CHARS
ET :: ON BREAK NEXT
380 CALL SCREEN(2)
390 FOR I=0 TO 14 :: CALL CO
LOR(I,2,2):: NEXT I
400 !@P+
410 DIM M(5),X(64),Y(64),EI$
(10),NA$(10)
420 !@P-
430 RESTORE 1920 :: FOR I=1
TO 64 :: READ Y1,X1 :: Y(I)=
Y1 :: X(I)=X1 :: NEXT I
440 CALL CHAR(104,RPT$("0",1
5)&"1",105,RPT$("0",12)&"FF0
0",106,RPT$("0",14)&"81")
450 CALL CHAR(107,RPT$("0",1
4)&"80",108,"0305"&RPT$("09"
,6),109,"C3A5"&RPT$("9",12))
460 CALL CHAR(110,"C0A090909
0909090",111,"090A0C0F",112,
"FF0000FF",113,"995A3CFF",11
4,"905030F0")
470 CALL CHAR(101,"0000EA85E
585E5",97,"304878CA2020203A"
,102,"00E384E8888483",99,"00
374282824232",115,"748484661
515E5")
480 M(1)=3 :: M(2)=4 :: M(3)
=5 :: M(4)=4 :: M(5)=4
490 CALL TASTA(M()):: CALL S
ONDER :: CALL ANZ(X(),Y())::
 DISPLAY AT(2,4):"TOUCH TYPI
NG TEACHER"
500 DISPLAY AT(21,2):"(C) 19
85 BY DIRK JUNGHANS"
510 DISPLAY AT(24,4):"PRESS
ANY KEY TO BEGIN"
520 CALL SCREEN(16)
530 FOR I=0 TO 8 :: CALL COL
OR(I,2,16):: NEXT I
540 FOR I=9 TO 12 :: CALL CO
LOR(I,5,16):: NEXT I
550 CALL KEY(0,K,S):: IF S=0
 THEN 550
560 CALL CLEAR
570 DISPLAY AT(1,1):"DJIS"
580 DISPLAY AT(5,1):"1 =>KEN
NENLERNEN MIT DER........TAS
TATUR"
590 DISPLAY AT(8,1):"2 =>WIE
```

```
1 JEDOCH MIT............VOR
GEGEBENEN ZEICHEN"
600 DISPLAY AT(11,1):"3 =>NA
CHSCHREIBEN VORHER........EI
NGEGEBENENER TEXTE"
610 DISPLAY AT(14,1):"4 =>SC
HREIBEN AUF ZEIT VON......FR
EIEN TEXTEN"
620 DISPLAY AT(17,1):"5 =>DR
UCKEN DER TASTATUR AUF....BI
LDSCHIRM"
630 DISPLAY AT(20,1):"6 =>EN
DE"
640 DISPLAY AT(24,4):"CHOOSE
 A KEY TO BEGIN"
650 CALL KEY(0,K,S):: IF S=0
 THEN 650
660 IF K<49 OR K>54 THEN 650
670 ON K-48 GOTO 740,1470,85
0,1570,1850,680
680 CALL CLEAR
690 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(10,2):"WOLLEN S
IE EIN NEUES"
700 DISPLAY AT(11,2):"PROGRA
MM LADEN ???"
710 DISPLAY AT(15,10):"(J/N)
" :: ACCEPT AT(15,16)BEEP VA
LIDATE("J","N")SIZE(1):JN$
720 IF JN$="N" THEN STOP
730 CALL CHARSET :: CALL CLE
AR :: RUN "CS1"
740 CALL CLEAR
750 CALL TASTA(M())
760 CALL SONDER
770 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(3,2):"NR. 1"
780 DISPLAY AT(20,2):"PRESS
FCTN 8 TO PRINT KEYS"
790 DISPLAY AT(21,2):"PRESS
FCTN 9 TO RETURN"
800 DISPLAY AT(24,2):"PRESS
NOW A KEY TO GO ON"
810 CALL KEY(0,K,S):: IF S=0
 THEN 810
820 DISPLAY AT(20,1):RPT$(""
,28):: DISPLAY AT(21,1):RPT$
("",28):: DISPLAY AT(24,1):R
PT$("",28)
830 CALL EING(Y(),X())
840 GOTO 560
850 CALL CLEAR :: ZAE=0
860 PRINT "JETZT KOENNEN SIE
 EINEN":"VON IHNEN EINGEGEBE
N TEXT":"NACHSCHREIBEN.ZUERS
T,MUESSEN"
870 PRINT "SIE EINEN 10 ZEIL
IGEN TEXT":"EINGEBEN.DANACH
ERSCHEINT"
880 PRINT "DER TEXT IMMER ZE
ILE FUER":"ZEILE AUF DEM BIL
DSCHIRM,"
890 PRINT "DIE SIE DANN NACH
SCHREIBEN":"MUESSEN.": : : :
900 PRINT TAB(2):"PRESS ANY
KEY TO CONTINUE"
910 DISPLAY AT(1,1):"DJIS" :
: DISPLAY AT(3,2):"NR.3"
920 CALL KEY(0,K,S):: IF S=0
 THEN 920 :: IF K=15 THEN 56
0
930 CALL CLEAR :: DISPLAY AT
(1,1):"DJIS"
940 DISPLAY AT(3,5):"EINGABE
 DES TEXTES"
950 DISPLAY AT(5,1):"START"
:: DISPLAY AT(17,25):"ENDE"
960 DISPLAY AT(12,3):"PRESS
ANY KEY TO START"
970 CALL KEY(0,K,S):: IF S=0
 THEN 970 :: IF K=15 THEN 56
0
980 DISPLAY AT(12,3):RPT$(""
,22)
990 FOR I=1 TO 10
1000 ACCEPT AT(I+5,1):EI$(I)
1010 NEXT I
1020 DISPLAY AT(20,1):"DIE E
INGABE IST BEENDET"
1030 DISPLAY AT(24,2):"PRESS
 ANY KEY TO CONTINUE"
1040 FOR I=1 TO 50 :: NEXT I
1050 CALL KEY(0,K,S):: IF S=
0 THEN 1050 :: IF K=15 THEN
560
1060 CALL CLEAR :: ZAE=0
1070 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1080 DISPLAY AT(10,2):"r DIE
SEN TEXT NACHSCHREIBEN"
1090 DISPLAY AT(6,1):RPT$("-
",28): :RPT$("-",28):: DISPL
AY AT(11,1):RPT$("=",28): :R
PT$("=",28)
1100 FOR I=1 TO 10
1110 DISPLAY AT(12,1)BEEP:EI
$(I)
1120 ACCEPT AT(7,1):NA$(I)
1130 NEXT I
1140 DISPLAY AT(15,6):"BITTE
 WARTEN !!!"
1150 FOR I=1 TO 10
1160 IF NA$(I)<>EI$(I)THEN G
OSUB 1430
1170 NEXT I
1180 DISPLAY AT(18,2):"SIE M
ACHTEN ";ZAE;" FEHLER"
1190 DISPLAY AT(19,2):"BEIM
ABSCHREIBEN"
1200 DISPLAY AT(22,1):"1 =>D
RUCKEN DER BEIDEN TEXTE"
1210 DISPLAY AT(23,1):"2 =>N
```

```
OCHMAL NACHSCHREIBEN"
1220 DISPLAY AT(24,1):"FCTN
9 =>ZURUECK ZUM MENUE"
1230 CALL KEY(0,K,S):: IF S=
0 THEN 1230 :: IF K=15 THEN
560
1240 IF K=49 THEN 1260
1250 IF K=50 THEN 1060 ELSE
1230
1260 CALL CLEAR
1270 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1280 DISPLAY AT(24,4):"PRESS
 A KEY TO PRINT"
1290 CALL KEY(0,K,S):: IF S=
0 THEN 1290 :: IF K=15 THEN
560
1300 DISPLAY AT(4,1):"EINGAB
ETEXT"
1310 FOR I=1 TO 10 :: DISPLA
Y AT(I+5,1):EI$(I):: NEXT I
1320 DISPLAY AT(19,1):RPT$("
-",28): :RPT$("-",28)
1330 DISPLAY AT(18,1):"NACHS
CHREIBE TEXT"
1340 DISPLAY AT(24,2):"PRESS
 ANY KEY TO CONTINUE"
1350 FOR I=1 TO 10
1360 DISPLAY AT(20,1):NA$(I)
1370 FOR I2=1 TO 50 :: NEXT
I2
1380 CALL KEY(0,K,S):: IF S=
0 THEN 1370 :: IF K=15 THEN
560
1390 NEXT I
1400 DISPLAY AT(23,1):"1 =>N
OCHMAL NACHSCHREIBEN"
1410 DISPLAY AT(24,1):"FCTN
9 =>ZURUECK ZUM MENUE"
1420 CALL KEY(0,K,S):: IF S=
0 THEN 1420 :: IF K=15 THEN
560 :: IF K=49 THEN 1060 ELS
E 1420
1430 FOR I1=1 TO 28
1440 IF SEG$(NA$(I),I1,1)<>S
EG$(EI$(I),I1,1)THEN ZAE=ZAE
+1
1450 NEXT I1
1460 RETURN
1470 CALL CLEAR
1480 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 2"
1490 CALL TASTA(M())
1500 CALL SONDER
1510 DISPLAY AT(20,2):"PRESS
 FCTN 9 TO RETURN"
1520 DISPLAY AT(24,2):"PRESS
 NOW A KEY TO GO ON"
1530 CALL KEY(0,K,S):: IF S=
0 THEN 1530
1540 DISPLAY AT(20,1):RPT$("
",28):: DISPLAY AT(24,1):RPT
$("",28)
1550 CALL ABFEING(Y(),X())
1560 GOTO 560
1570 CALL CLEAR
1580 TI=0
1590 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 4"
1600 DISPLAY AT(6,1):"BITTE
DRUECKEN SIE EINE":"TASTE,DA
NN BEGINNT DER":"ZAEHLER ZU
ZAEHLEN"
1610 DISPLAY AT(9,1):"UND SI
E KOENNEN BEGINNEN":"MIT DEM
 SCHREIBEN.":"VIEL GLUECK !"
1620 DISPLAY AT(13,1):"SIE M
UESSEN 168 ZEICHEN":"SCHREIB
EN.DIE SCHWARZEN"
1630 DISPLAY AT(15,1):"KAEST
CHEN WERDEN AM ENDE":"NICHT
MITGEZAEHLT."
1640 DISPLAY AT(17,1):"SIE K
OENNEN AUCH KEINE":"KORREKTU
REN AM TEXT VOR-"
1650 DISPLAY AT(19,1):"NEHME
N.SCHREIBEN SIE DES-":"HALB
BEI EINEM FEHLER":"WEITER."
1660 DISPLAY AT(24,3):"PRESS
 ANY KEY TO START"
1670 CALL KEY(0,K,S):: IF S=
0 THEN 1670 :: CALL CLEAR
1680 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 4"
1690 DISPLAY AT(4,1):"START"
 :: DISPLAY AT(11,25):"ENDE"
1700 S=0 :: SE=5 :: WA=3
1710 CALL KEY(0,K,S):: TI=TI
+1 :: IF S=0 THEN 1710
1720 CALL HCHAR(SE,WA,K):: C
ALL HCHAR(SE,WA+1,30)
1730 WA=WA+1
1740 IF WA>30 THEN WA=3 :: S
E=SE+1 ELSE 1750
1750 IF NOT SE>10 THEN GOTO
1710
1760 DISPLAY AT(17,2):"SIE B
RAUCHTEN FUER 168"
1770 DISPLAY AT(18,2):"ZEICH
EN ";TI
1780 DISPLAY AT(19,2):"ZEITE
INHEITEN."
1790 DISPLAY AT(20,2):"NICHT
 SCHLECHT !!!"
1800 DISPLAY AT(23,2):"1 =>N
OCH EINMAL"
1810 DISPLAY AT(24,2):"FCTN
9 =>ZURUECK ZUM MENUE"
1820 FOR I=1 TO 100 :: NEXT
I
1830 CALL KEY(0,K,S):: IF S=
0 THEN 1820 :: IF K=49 THEN
```

```
1570 :: IF K<>15 THEN 1830
1840 GOTO 560
1850 CALL CLEAR
1860 DISPLAY AT(1,1):"DJIS"
:: DISPLAY AT(3,2):"NR. 5"
1870 CALL TASTA(M())
1880 CALL SONDER
1890 CALL ANZ(X(),Y())
1900 DISPLAY AT(24,4):"PRESS
 A KEY TO RETURN"
1910 CALL KEY(0,K,S):: IF S=
0 THEN 1910 :: GOTO 560
1920 DATA 6,4,9,23,6,8,6,10,
6,12,6,16,9,21,6,20,6,22,6,1
8,6,24,15,21,9,25,15,23,9,25
,6,22
1930 DATA 6,4,6,6,6,8,6,10,6
,12,6,14,6,16,6,18,6,20,12,2
4,12,24,15,21,6,24,15,23
1940 DATA 9,19,6,6,12,6,15,1
5,15,11,12,10,9,9,12,12,12,1
4,12,16,9,19,12,18,12,20
1950 DATA 12,22,15,19,15,17,
9,21,9,23,9,5,9,11,12,8,9,13
,9,17,15,13,9,7,15,9,9,15,15
,7
1960 DATA 9,11,15,7,9,13,6,1
4,9,17,15,11
1970 !@P+
1980 SUB TASTA(M()):: DJ=5
1990 FOR I=1 TO 5
2000 DISPLAY AT(DJ,M(I)):"h"
&RPT$("ij",10)&"ik"
2010 DISPLAY AT(DJ+1,M(I)):"
l"&RPT$(" m",10)&" n"
2020 DISPLAY AT(DJ+2,M(I)):"
o"&RPT$("pq",10)&"pr"
2030 DJ=DJ+3
2040 NEXT I
2050 CALL HCHAR(17,11,105,15
):: CALL HCHAR(19,11,112,15)
:: CALL HCHAR(18,11,32,15)
2060 CALL HCHAR(14,28,105)::
 CALL HCHAR(14,29,107):: CAL
L HCHAR(15,28,32):: CALL HCH
AR(15,29,110)
2070 CALL HCHAR(16,28,112)::
 CALL HCHAR(16,29,114)
2080 SUBEND
2090 SUB ABFEING(Y(),X())
2100 DD=INT((96-32+1)*RND)+3
2
2110 IF DD=32 THEN GOSUB 226
0 :: GOTO 2130 ELSE 2120
2120 CALL HCHAR(Y(DD-32),X(D
D-32)+2,DD)
2130 CALL KEY(0,K,S):: IF S=
0 THEN 2130
2140 IF K=15 THEN SUBEXIT
2150 IF K>96 OR K<32 THEN CA
LL SOUND(100,300,3,-7,3):: G
```

```
OTO 2130 :: ELSE 2160
2160 IF K<>DD AND K=32 THEN
GOSUB 2270 :: GOTO 2240 ELSE
 IF K<>DD THEN 2210
2170 CALL SOUND(100,392,0)::
 CALL SOUND(100,392,0,330,0)
:: CALL SOUND(100,392,0,330,
0):: CALL SOUND(200,440,0,39
2,0)
2180 CALL SOUND(200,392,0,33
0,0)
2190 IF DD<>32 THEN 2200 ::
CALL HCHAR(18,15,32,5):: GOT
O 2100
2200 CALL HCHAR(Y(DD-32),X(D
D-32)+2,32):: GOTO 2100
2210 CALL HCHAR(Y(K-32),X(K-
32)+2,K)
2220 FOR I=1 TO 30 :: NEXT I
2230 CALL HCHAR(Y(K-32),X(K-
32)+2,32):: CALL HCHAR(Y(DD-
32),X(DD-32)+2,DD)
2240 CALL SOUND(100,220,0)::
 CALL SOUND(100,175,0):: CAL
L SOUND(200,-3,0)
2250 GOTO 2130
2260 CALL HCHAR(18,15,30,5):
: RETURN
2270 CALL HCHAR(18,15,30,5)
2280 FOR I=1 TO 30 :: NEXT I
2290 CALL HCHAR(18,15,32,5):
: RETURN
2300 SUBEND
2310 SUB ANZ(X(),Y())
2320 FOR I=12 TO 59 :: IF I+
32>90 OR I+32<44 OR I+32=45
OR I+32=58 OR I+32=60 OR I+3
2=62 OR I+32=63 OR I+32=64 T
HEN 2340
2330 CALL HCHAR(Y(I),X(I)+2,
I+32)
2340 NEXT I
2350 SUBEND
2360 SUB SONDER
2370 CALL HCHAR(15,27,115)::
 CALL HCHAR(12,28,101)
2380 CALL HCHAR(15,7,115)::
CALL HCHAR(18,7,97):: CALL H
CHAR(18,9,99):: CALL HCHAR(1
8,27,102)
2390 SUBEND
2400 SUB EING(Y(),X())
2410 CALL KEY(0,K,S):: IF S=
0 THEN 2410 :: IF K<>6 THEN
2440
2420 CALL ANZ(X(),Y())
2430 GOSUB 2510 :: GOTO 2410
2440 IF K=15 THEN SUBEXIT
2450 IF K<>32 THEN 2460 ELSE
 CALL HCHAR(18,15,30,5):: GO
TO 2500
```

```
2460 IF K>96 OR K<33 THEN CA
LL SOUND(100,300,3,-7,3):: G
OTO 2410 ELSE 2470
2470 CALL HCHAR(Y(K-32),X(K-
32)+2,K)
2480 FOR I=1 TO 30 :: NEXT I
 :: CALL HCHAR(Y(K-32),X(K-3
2)+2,32)
2490 GOTO 2410
2500 CALL HCHAR(18,15,32,5):
: GOTO 2410
2510 DISPLAY AT(24,3):"PRESS
 A KEY TO ERASE KEYS"
2520 CALL KEY(0,K,S):: IF S=
0 THEN 2520 :: DISPLAY AT(24
,3):RPT$(" ",25)
2530 FOR I=12 TO 59 :: CALL
HCHAR(Y(I),X(I)+2,32):: NEXT
 I :: RETURN
2540 SUBEND
2550 !@P-
2560 END
```

DER TI ALS SCHREIB-MASCHINEN LEHRER

Fortsetzung von Seite 128

wie in allen anderen Unterprogrammen lediglich FCTN 9 drücken.

Wenn Sie 3 drücken, gelangen Sie in das Unterprogramm „Nachschreiben vorher eingegebener Texte“. Nun können Sie einen vorher eingegebenen Text nachschreiben. Dazu erscheint zuerst einmal eine kurze Erklärung auf dem Bildschirm, was in diesem Unterprogramm gemacht werden muß.

Im oberen Teil des Bildschirms wird jeweils eine Zeile angezeigt, die Sie dann in der Zeile etwas darunter nachschreiben müssen. Auch hier müssen Sie wieder spätestens am Ende der Zeile ENTER drücken. Wenn Sie dies gemacht haben, wird die nächste Zeile eingeblendet, die Sie dann nachschreiben müssen. Haben Sie alle 10 Zeilen nachgeschrieben, müssen Sie dann etwas warten, weil der Rechner dann überprüft, ob Sie Fehler beim Abschreiben gemacht haben. Die Anzahl der Fehler wird dann angezeigt. Danach können Sie dann wieder wählen, ob Sie mit FCTN 9 ins Menue zurück wollen, ob Sie mit 2 den Text nochmal nachschreiben wollen oder ob Sie mit 1 den nachgeschriebenen und den nachzuschreibenden Text angezeigt haben wollen.

Wenn Sie 4 drücken, können Sie messen, wie schnell Ihre Schreibgeschwindigkeit ist. Zuerst erscheint auch hier wieder eine kurze Einleitung, was in diesem Unterprogramm zu machen ist. Wenn Sie sich die kurze Einleitung durchgelesen haben, müssen Sie eine Taste drücken, damit das Programm fortfährt.

Wenn Sie 5 im Hauptmenü gewählt haben, druckt der Computer Ihnen noch einmal die TI-Tastatur auf dem Bildschirm aus, damit Sie noch einmal einen Überblick bekommen, wo die einzelnen Tasten auf der Tastatur liegen. Das können Sie natürlich auch auf der richtigen Tastatur machen, aber erstens sollen Sie bei dem Programm nie auf die Tastatur sehen und zweitens gewöhnen Sie sich dann besser an die Bildschirmtastatur, wenn Sie z.B. mit Nr. 1 des Hauptmenüs arbeiten wollen.

Falls Sie 6 drücken, können Sie das Programm beenden. Zuerst werden Sie aber noch gefragt, ob Sie ein neues Programm laden wollen? Wenn Sie ja drücken, gibt der Rechner automatisch die Anweisungen auf dem Bildschirm aus, um ein neues Programm zu laden (z.B. Rewind Cassette Tape). Wenn Sie Besitzer einer Diskettenstation sind, müssen Sie nur die Zeile 780 ändern. Wenn Sie aber statt J, N drücken, wird das Programm beendet.

Variablenliste:
STRINGS:
JN$ – String für die JA/NEIN-Abfrage in Nummer 6 des Hauptmenüs
EI$ ( ) – String (Datenfeld) für den nachzuschreibenden Text in „Nachschreiben vorher eingegebener Texte“
NA$ ( ) – String (Datenfeld) für den nachgeschriebenen Text in „Nachschreiben vorher eingegebener Texte“

VARIABLE:
I, I1, I2 – Variable für FOR-NEXT Schleifen
ZAE – Variable für die Anzahl der Fehler in „Nachschreiben vorher eingegebener Texte“
TI – Variable für die gebrauchten Zeiteinheiten in „Schreiben auf Zeit von freien Texten“
SE – Variable für den senkrechten Standpunkt des Cursors in „Schreiben auf Zeit von freien Texten“
WA – Variable für den waagerechten Standpunkt des Cursors in „Schreiben auf Zeit von freien Texten“
K, S – Variable für die CALL KEY-Abfrage
DJ – Wert für den waagerechten Standpunkt der Tasten auf der Bildschirmtastatur
DD – Zufallswert für die Auswahl der Symbole in Nr. 2 des Hauptmenüs
X1, Y1 – Werte für die „Data-Auslesung“
M ( ) – Variable (Datenfeld) für den senkrechten Standpunkt der Tasten auf der Bildschirmtastatur
X ( ) – Variable (Datenfeld) für den X-Wert der einzelnen Symbole auf der Bildschirmtastatur
Y ( ) – Variable (Datenfeld) für den Y-Wert der einzelnen Symbole auf der Bildschirmtastatur

Dirk Junghans

★

SPRITE-KONSTRUKTEUR

Wie hat man früher Sprites auf dem TI-99/4A erstellt? Man nahm Zettel und Bleistift und zeichnete zunächst einen Raster mit 16 mal 16 Feldern auf das Papier. Dann mußte man sich zunächst das Sprite gründlich überlegen, denn probieren war viel zuviel Arbeit. Schließlich kritzelte man das Sprite in den Raster und wenn es dann doch nicht den Anforderungen entsprach, blieb einem nichts anderes übrig als der Griff zum Radiergummi. Der Fehler wurde also nun ausradiert, die Rasterlinien nachgezogen und die Prozedur von neuem wiederholt.

Dann kam der peinlichste Teil der Arbeit: Mühevoll mußte man in Zusammenarbeit mit dem Bedienungsbuch die Hexcodes raussuchen — dauerndes Vergleichen mit der Zeichnung und dem Anleitungsbuch. Hatte man nun schließlich tatsächlich die sechzehn richtigen Codes herausgeschrieben, so sah man sich zunächst einmal das neue Sprite am Bildschirm an. Dazu mußte die Kolonne natürlich zuerst eingegeben werden, ein eigenes Programm dafür geschrieben werden, usw.

Jeder TI-99-Besitzer kennt diese Arbeiten nur zu gut. Oftmals verdirbt nur der Gedanke an sie dem Programmierer die Lust, ein neues Programm zu machen.

Wie kann man nun Sprites auf dem TI-99/4A erstellen? Ganz einfach: Mit dem Spriteconstruction-Set. Legen Sie Ihre Zettel, Bleistifte, Radiergummi und Bedienungsanleitungen ruhig zur Seite. Mit dem Spriteconstruction-Set wird das Erstellen von Sprites ein Kinderspiel. Sie malen die Form des Sprites einfach auf den Bildschirm, lassen es vom Computer in jeder beliebigen Darstellungsart bearbeiten, können sich das Ganze auch in „Natura" anschauen und bei Mißfallen nach Belieben Änderungen vornehmen. Auch die lästige Arbeit der Hex-Code-Bestimmung übernimmt für Sie nun der Computer. Und als Krönung haben Sie die Möglichkeit, bis zu 18 Sprites „aufzuheben", um sie später nochmals anzusehen, oder mit anderen Sprites zu vergleichen.

Spriteconstruction-Set

Ein extrem leistungsfähiges Hilfsprogramm. Mit dem Sprite-Construction Set können Sprites in Sekundenschnelle erstellt werden. Was bisher bie der Entwicklung von eigenen Progammen die anstrengendste und gefürchtetste Arbeit war, wird nun zur unterhaltsamen Beschäftigung. Ohne Bleistift, Zettel und Radiergummi malen Sie Ihre Sprites einfach auf den Bildschirm, lassen sie vom Computer drehen oder wenden, sehen sich Ihre Meisterwerke in Natura an und lassen auch den Hexcode vom Computer berechnen.

Die Möglichkeiten des Programms sind so umfangreich, daß sie hier gar nicht alle angeführt werden können. Eine sechsseitige Programmdokumentation liegt diesem Schreiben bei. Jeder TI-User wird das Sprite-Constructionset sicher sehr bald zu schätzen wissen!

Nach dem Starten des Programms mit RUN bildet der Computer zunächst das Zeichenfeld am Bildschirm ab und gleich danach das Hauptmenue. Neben der ersten Auswahlmöglichkeit des Hauptmenues befindet sich ein weißer Pfeil.

Sie können diesen Pfeil bewegen, indem Sie den Joystick in die entsprechende Richtung drükken. Der Pfeil wesit dann auf den nächsten Namen der Liste.

Beachten Sie die anderen Namen der Liste zunächst nicht, und führen Sie den Pfeil neben das Wort „WORKING".

Wenn Sie nun den Actionknopf Ihres Joysticks drücken, gelangen Sie in das Arbeitssystem, in dem Sie Ihre Sprites erstellen können. Die anderen Teile des Hauptmenues werden für Sie erst interessant, wenn Sie bereits erstellte Sprites verwalten (Speichern, aufrufen) wollen.

Im Arbeitsmodus wird zunächst das Arbeitsmenue aufgelistet. Es besteht aus folgenden Auswahlmöglichkeiten:

SYMMETRY, INVERT, MIRROR, ROTATION HEX IN, HEX OUT, LOOK und READY

Beachten Sie vorläufig diese Auswahlmöglichkeiten noch nicht und drükken Sie den Joystick nach rechts.

Sie werden bemerkt haben, daß der weiße Pfeil neben dem Wort SYMMETRY des Menues verschwunden ist. Das bedeutet, daß Sie das Menue verlassen haben, und sich jetzt im Arbeitsmodus befinden.

Im Zeichenfeld befindet sich nun ein Zielkreuz, das Sie in alle Richtungen bewegen können. Wenn Sie den Actioknopf drücken, können Sie nun Punkte Feld fahren, auf dem sich bereits ein Punkt befindet und Sie abermals den Actionknopf drücken, wird er wieder gelöscht.

Auf diese Art und Weise können Sie nun völlig problemlos Ihr Sprite „malen". Wenn Sie längere Linien ziehen müssen, oder Flächen ausmalen wollen, dann müssen Sie bei gedrücktem Feuerknopf in die entsprechende Richtung drücken und schon wird eine Linie gezogen. Jede Aktion Ihrerseits wird mit einem akustischen Signal untermalt.

Sie können den Arbeitsmodus wieder verlassen, indem Sie mit dem Zielkreuz über den Zeichenfeldrand hinaussteuern. Sobald dies geschehen ist, verschwindet das Zielkreuz und Sie befinden sich wieder im Auswahlmodus des Menues.

Das Arbeitsmenue

Das Arbeitsmenue ist imstande, Sie bei der Erstellung Ihrer Sprites wesentlich zu unterstützen. Sie können ein Programm des Arbeitsmenues auf-

rufen, indem Sie den Pfeil neben das gewünschte Wort steuern und dann den Actionknopf drücken. Ein weißer Punkt neben dem Wort zeigt Ihnen dann an, daß das Programm aktiviert wurde.

Symmetry:

Das Symmetry-Programm hilft Ihnen, wenn sie symmetrische Figuren erstellen wollen. Sie müssen dann, je nach Art der Symmetrie, nur ein Viertel oder die Hälfte der Zeichenfläche „bemalen" und lassen dann den Computer für Sie weitermalen.

Wenn Sie im Arbeitsmenue SYMMETRY gewählt haben, gelangen Sie zunächst in ein kleines Nebenmenue.

Hier stehen H-AXIAL, V-AXIAL und CIRCULAR zur Auswahl.

Neben den Wörtern ist ein Symbol abgebildet, welches veranschaulicht, welches Viertel bzw. welche Hälfte Sie bemalen müssen.

Circular:

Wenn Sie dieses Programm wählen, setzt der Computer Zirkularsymmetrie voraus. Das linke obere Veirtel wird dann zirkularsymmetrisch vervielfacht.

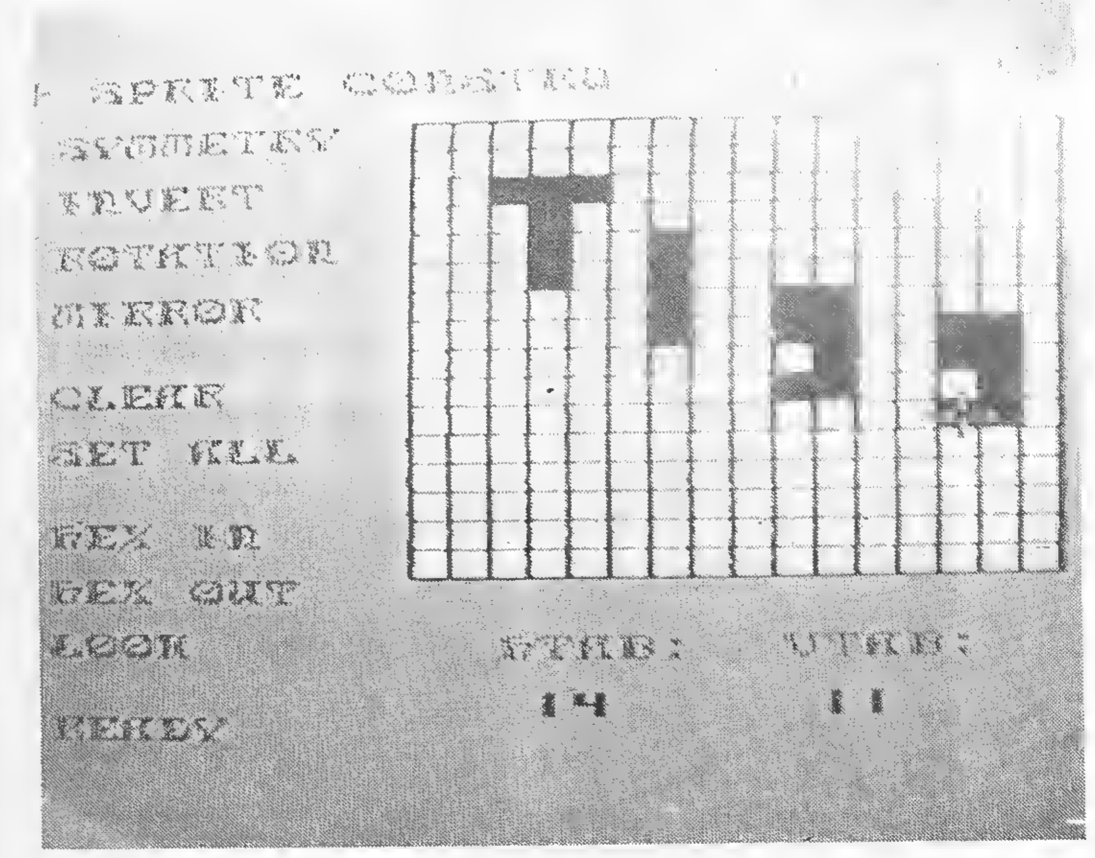
Dieses Programm eignet sich ideal zur Gestaltung von Spielbällen oder ähnlichen symmetrischen Figuren. Sie müssen lediglich ein Viertel des Balles zeichnen, steigen dann in die Programme SYMMETRY, CIRCULAR ein, und der Computer zeichnet das Gebilde fertig.

H-Axial:

Das Programm H-Axial setzt die horizontale Mittelachse als Symmetrieachse voraus. Zeichnen müssen Sie, bevor Sie dieses Programm wählen, daher die obere Hälfte des Sprites. Die untere Hälfte zeichnet dann der Computer.

V-Axial:

Das Programm V-Axial setzt die vertikale Mittelachse als Symmetrieachse voraus. Hier muß die linke Hälfte des Sprites gezeichnet sein, damit der Computer die andere Hälfte erstellen kann.



Invert:

Das INVERT-Programm invertiert ein fertig gezeichnetes Sprite. Das heißt, alle schwarzen Punkte im Zeichenfeld werden weiß und weiße Punkte werden schwarz. Wie bei allen anderen Hilfsprogrammen des Arbeitsmenues können Sie nach Verwendung des Programms wieder in den Arbeitsmodus einsteigen und Veränderungen vornehmen.

Mirror:

Das Mirror Programm spiegelt ein bereits fertiges Sprite. Auch hier gelangen Sie in ein Nebenmenue, welches Sie fragt, ob HORIZONTAL oder VERTIKAL gespiegelt werden soll.

Horizontal:

Spiegelt Horizontal. Ideal bei der Erstellung von Spielfiguren! Sie zeichnen zum Beispiel nur die Figur, die nach links geht, spiegeln diese dann horizontal und schon haben Sie die rechte Figur.

Vertikal:

Genauso wie HORIZONTAL. Es wird allerdings vertikal gespiegelt.

Rotation:

Wenn Sie das Rotation-Programm gewählt haben, taucht unterhalb des Zeichenfeldes ein neues Menue auf, welches sich erkundigt, ob Sie das Sprite nach links oder nach rechts rotieren lassen wollen.

Wenn Sie nun links gewählt haben, führt das Sprite eine 90 Grad-Dehnung nach links aus. Wählen Sie rechts, so geschieht derselbe Vorgang nach rechts.

Durch öftere Anwendung dieses Programmes kann man ein Sprite in jeder beliebigen Richtung darstellen. Bei der Definition von Spielfiguren ist dieses Programm unentbehrlich.

So wie die vorangegangenen vier Programme ist auch das nun folgende fünfte Programm des Arbeitsmenues bei der Erstellung von Sprites behilflich:

Bitte lesen Sie weiter auf Seite 139

```
1Ø ! **********************
11 ! *                    *
12 ! * SPRITE-CONSTRUCTION*
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *    Peter Wenzel    *
17 ! *                    *
18 ! *                    *
19 ! * Benoetigte Geraete *
2Ø ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     913Ø Bytes     *
28 ! *                    *
29 ! **********************
3Ø !
4Ø ! Bei diesem Programm hab
en wir auf die Hilfspunkte v
erzichtet ! (Die Redaktion)
19Ø CALL CLEAR :: ON WARNING
 NEXT :: DIM H$(16),D$(1),SA
VE$(18):: FOR I=1 TO 16 :: R
EAD H$(I):: NEXT I
2ØØ DATA ((((,((((),(()(,(())
,()((,()(),())(,())),)(((,)(
(),)()(,)()),))((,))(),)))(,
))))
21Ø DATA FF,Ø1,Ø1Ø1Ø1Ø1Ø1Ø1Ø
1Ø1,FFØ1Ø1Ø1Ø1Ø1Ø1Ø1,FFFFFFF
FFFFFFFFF,Ø,FFF1F1F1F1F1F1FF
,FFFFFFFF818181FF,FFF1F1F181
8181FF,Ø,Ø
22Ø DATA ØØFCCCCCCCCCCCFC,ØØ
3Ø3Ø3Ø3Ø3Ø3Ø3,ØØFCØCØCFCCØCØ
FC,ØØFCØCØC7CØCØCFC,ØØCCCCCC
FCØCØCØC,ØØFCCØCØFCØCØCFC,ØØ
CØCØCØFCCCCCFC
23Ø DATA ØØFCØCØCØCØCØCØC,ØØ
FCCCCCFCCCCCFC,ØØFCCCCCFCØCØ
CØC,ØØFCCCCCFCCCCCCC,ØØFØCCC
CFØCCCCF,ØØFCCØCØCØCØCØFC,ØØ
FØCCCCCCCCCCCF,ØØFCCØCØF8CØCØ
FC
24Ø DATA ØØFCCØCØF8CØCØC,3C4
299A1A199423C,ØØ3F666C746466
EF,ØØ7E313B3633337E,ØØ3E6341
4Ø6Ø3F1E,ØØ7E3211191 13F7E,ØØ
7F33Ø3C38317F,ØØ7F723239383
Ø78
25Ø DATA ØØ3E66485Ø4F663E,ØØ
7C6Ø3F313232 7AØ3,ØØ7C38181C1
A183C2,ØØ3F13Ø6Ø6Ø6263F,ØØ7B
72363C363673,ØØ783Ø3Ø3Ø72637
F
26Ø DATA ØØ7F2B2A49494963,ØØ
7C76323A323477,ØØ3E634559617
F3E,ØØ7E33313F3E3Ø78,ØØ3E634
559617F3EØ7,ØØ7F313A34323179
,ØØ3E413CØ6634F3F
27Ø DATA ØØ7F492AØ8Ø8Ø81C,ØØ
6F262666 6E7E37,ØØ7362223232 1
EØC,ØØ7722224149 4963,ØØ79321
4Ø814264F,ØØ7341623418183C,Ø
Ø7F4326ØC1A317F,1Ø1Ø1ØEE1Ø1Ø
1
28Ø DATA ØØØØ181 8ØØØØ1818,ØØ
Ø8ØC7E7F7EØCØ8
29Ø FOR I=37 TO 93 :: READ I
$ :: CALL CHAR(I,I$):: NEXT
I
3ØØ CALL COLOR(1,2,1,2,2,16,
3,5,1,4,5,1,5,9,1,6,9,1,7,9,
1,8,9,1,9,3,1,1Ø,16,1,12,2,1
6)
31Ø CALL CLEAR :: CALL SCREE
N(8):: DISPLAY AT(1,1):"[[ S
PRITE CONSTRUCTIONSET [[" ::
 CALL SET
32Ø RESTORE 124Ø :: CALL DIR
 :: DISPLAY AT(2Ø,13):"@ BY
PETER WENZL"
33Ø Z1=5 :: S1=3 :: GOTO 35Ø
34Ø CALL S :: CALL C :: DISP
LAY AT(2Ø,13):" [[[[ READY [
[[[" :: Z1=Z :: S1=S
35Ø GOSUB 116Ø :: IF X=4 THE
N 35Ø
36Ø ON Z-4 GOTO 5ØØ,1ØØ,45Ø,
1ØØ,1ØØ,1ØØ,1ØØ,37Ø,1ØØ,39Ø,
1ØØ,1ØØ,1ØØ,1ØØ,4ØØ,1ØØ,1ØØ,
1ØØ,1ØØ,1ØØ
37Ø GOSUB 93Ø
38Ø CALL KEY(1,K,Q):: IF K=1
8 THEN Z=12 :: S=3 :: GOTO 3
4Ø ELSE 38Ø
39Ø GOSUB 1ØØØ :: Z=14 :: GO
TO 34Ø
4ØØ IF HCV=Ø THEN CALL SOUND
(3ØØØ,44Ø,Ø):: DISPLAY AT(2Ø
,13):"CAN NOT BE SAVED" :: D
ISPLAY AT(22,13):"USE HEXOUT
 BEVOR" :: CALL SOUND(1,44Ø,
Ø):: Z=19 :: S=3 :: GOTO 34Ø
41Ø IF SN>17 THEN CALL SOUND
(3ØØØ,44Ø,Ø):: DISPLAY AT(2Ø
,13):"SORRY\" :: DISPLAY AT(
22,13):"MEMORY FULL" :: CALL
 SOUND(1,44Ø,Ø):: Z=19 :: S=
3 :: GOTO 34Ø
42Ø SN=SN+1 :: CALL C :: CAL
L S :: DISPLAY AT(2Ø,13):"SP
RITENAME\" :: ACCEPT AT(22,1
3)VALIDATE(UALPHA):I$ :: CAL
L S2 :: SAVE$(SN)=HEX$&HR$&I
$
43Ø CALL S :: DISPLAY AT(24,
13):"NUMBER";SN;"SAVED"
44Ø CALL KEY(1,K,Q):: IF K=1
8 THEN Z=19 :: S=3 :: GOTO 3
4Ø ELSE 44Ø
```

```
450 IF SN=Ø THEN Z=7 :: S=3
:: GOTO 34Ø :: ELSE DISPLAY
AT(2Ø,13):"SPRITENUMBER\" ::
 CALL S :: ACCEPT AT(2Ø,27):
Q :: IF Q<1 OR Q>SN THEN 45Ø
46Ø CALL S2 :: DISPLAY AT(22
,13):"SPRITENAME\" :: DISPLA
Y AT(24,13):SEG$(SAVE$(Q),12
9,16):: HEX$=SEG$(SAVE$(Q),1
,64):: HR$=SEG$(SAVE$(Q),65,
64)
47Ø HCV=1 :: GOSUB 89Ø :: CA
LL S2 :: CALL S2 :: CALL S2
48Ø CALL KEY(1,K,Q):: IF Q=Ø
 THEN 48Ø ELSE CALL S :: Z=7
 :: S=3 :: GOTO 34Ø
49Ø CALL S :: CALL C :: DISP
LAY AT(2Ø,13):" [[[[ READY [
[[[" :: Z1=Z :: S1=S :: GOTO
 52Ø
5ØØ RESTORE 125Ø :: CALL DIR
51Ø CALL C :: CALL DELSPRITE
(ALL):: Z1,S1=3
52Ø GOSUB 116Ø :: IF X=4 THE
N 11ØØ
53Ø IF Z=24 THEN 32Ø
54Ø ON Z-2 GOTO 55Ø,1ØØ,62Ø,
1ØØ,65Ø,1ØØ,72Ø,1ØØ,1ØØ,77Ø,
1ØØ,78Ø,1ØØ,1ØØ,79Ø,1ØØ,91Ø,
1ØØ,99Ø
55Ø HCV=Ø :: CALL SOUND(2ØØ,
88Ø,9,77Ø,9,66Ø,9):: DISPLAY
 AT(2Ø,14):"' AXIAL\ H ] ,"
:: DISPLAY AT(22,14):"' AXIA
L\ V ] +" :: DISPLAY AT(24,1
4):"' CIRCULAR ] -"
56Ø Z1=2Ø :: S1=16 :: GOSUB
116Ø :: IF X=4 THEN 56Ø
57Ø IF Z=2Ø THEN 6ØØ ELSE IF
 Z=22 THEN 61Ø
58Ø FOR I=3 TO 1Ø :: CALL SO
UND(-425Ø,-1,9):: FOR Q=15 T
O 22
59Ø CALL GCHAR(I,Q,C1):: CAL
L HCHAR(I,45-Q,C1):: CALL HC
HAR(21-I,45-Q,C1):: CALL HCH
AR(21-I,Q,C1):: NEXT Q :: NE
XT I :: GOTO 51Ø
6ØØ FOR I=3 TO 1Ø :: CALL SO
UND(-425Ø,-2,9):: FOR Q=15 T
O 3Ø :: CALL GCHAR(I,Q,C1)::
 CALL HCHAR(21-I,Q,C1):: NEX
T Q :: NEXT I :: GOTO 51Ø
61Ø FOR I=3 TO 18 :: CALL SO
UND(-425Ø,-3,Ø):: FOR Q=15 T
O 22 :: CALL GCHAR(I,Q,C1)::
 CALL HCHAR(I,45-Q,C1):: NEX
T Q :: NEXT I :: GOTO 51Ø
62Ø HCV=Ø :: DISPLAY AT(2Ø,1
3):" [[ INVERTING [[" :: FOR
 I=3 TO 18 :: CALL SOUND(-2Ø
ØØ,-4,9):: FOR Q=15 TO 3Ø ::
 CALL GCHAR(I,Q,K)
63Ø IF K=4Ø THEN CALL HCHAR(
I,Q,41)ELSE CALL HCHAR(I,Q,4
Ø)
64Ø NEXT Q :: NEXT I :: GOTO
 49Ø
65Ø HCV=Ø :: D$(Ø),D$(1)=""
:: CALL SOUND(2ØØ,88Ø,9,77Ø,
9,66Ø,9):: DISPLAY AT(22,13)
:"' LEFT" :: DISPLAY AT(24,1
3):"' RIGHT" :: Z1=22 :: S1=
15
66Ø GOSUB 116Ø :: IF X=4 THE
N 66Ø ELSE IF Z=22 THEN 7ØØ
67Ø DISPLAY AT(2Ø,13):"[[[ R
OTATION [[["
68Ø FOR A=Ø TO 1 :: FOR I=15
 TO 22 :: CALL SOUND(-4ØØØ,-
7,9):: FOR Q=18 TO 3 STEP -1
 :: CALL GCHAR(Q,I+A*8,C1)::
 D$(A)=D$(A)&CHR$(C1):: NEXT
 Q :: NEXT I :: NEXT A
69Ø GOSUB 126Ø :: Z=7 :: S=3
 :: GOTO 49Ø
7ØØ DISPLAY AT(2Ø,13):"[[[ R
OTATION [[["
71Ø FOR A=Ø TO 1 :: FOR I=3Ø
 TO 23 STEP -1 :: CALL SOUND
(-4ØØØ,-6,9):: FOR Q=3 TO 18
 :: CALL GCHAR(Q,I-A*8,C1)::
 D$(A)=D$(A)&CHR$(C1):: NEXT
 Q :: NEXT I :: NEXT A :: GO
TO 69Ø
72Ø CALL CHAR(12Ø,"Ø32466FFF
F6624ØØ183C7E18187E3C18")::
HCV=Ø :: CALL SOUND(2ØØ,88Ø,
9,77Ø,9,66Ø,9):: DISPLAY AT(
22,13):"' HORIZONTAL ] x"
73Ø DISPLAY AT(24,13):"' VER
TICAL   ] y" :: Z1=22 :: S1=
15 :: GOSUB 116Ø :: DISPLAY
AT(2Ø,13):"[[[[ MIRROR [[[["
 :: IF Z=22 THEN 76Ø
74Ø FOR I=15 TO 3Ø :: CALL S
OUND(-4ØØØ,-3,9):: FOR Q=1 T
O 8 :: CALL GCHAR(Q+2,I,C1):
: CALL GCHAR(19-Q,I,C2):: CA
LL HCHAR(Q+2,I,C2):: CALL HC
HAR(19-Q,I,C1):: NEXT Q :: N
EXT I
75Ø CALL S :: CALL C :: Z=9
:: S=3 :: GOTO 49Ø
76Ø FOR I=3 TO 18 :: CALL SO
UND(-4ØØØ,-2,9):: FOR Q=1 TO
 8 :: CALL GCHAR(I,Q+14,C1):
: CALL GCHAR(I,31-Q,C2):: CA
LL HCHAR(I,Q+14,C2):: CALL H
CHAR(I,31-Q,C1):: NEXT Q ::
NEXT I :: GOTO 75Ø
77Ø HCV=Ø :: CALL SET :: GOT
```

```
O 490
780 HCV=0 :: FOR I=3 TO 18 :
: CALL SOUND(-10,-6,0):: CAL
L HCHAR(I,15,41,16):: NEXT I
 :: GOTO 490
790 CALL CHAR(120,"FFF1F1F18
18181FFFF818181F1F1F1FFFF8F8
F8F818181FFFF8181818F8F8FFF"
):: CALL S :: DISPLAY AT(20,
13):"INSERT HEXCODE\" :: HR$
,HEX$=""
800 HCV=1 :: FOR I=1 TO 4 ::
 DISPLAY AT(22,13):"PARTNUMB
ER";I;"] ";CHR$(119+I)
810 S=15
820 CALL HCHAR(24,S,105)
830 CALL KEY(3,K,Q):: IF Q=0
 THEN 830 ELSE IF K<48 OR K>
57 THEN IF K<65 OR K>70 THEN
 CALL SOUND(100,-1,0):: GOTO
 830
840 CALL SOUND(12,999,8,-7,8
):: IF K>57 THEN K=K-7
850 CALL HCHAR(24,S,K):: HEX
$=HEX$&CHR$(K):: S=S+1 :: IF
 K>57 THEN HR$=HR$&CHR$(K+7)
ELSE HR$=HR$&CHR$(K)
860 IF S<31 THEN 820 ELSE CA
LL SOUND(50,-5,0):: CALL SOU
ND(50,-6,0):: CALL SOUND(50,
-3,0):: CALL HCHAR(24,15,32,
17)
870 NEXT I
880 GOSUB 890 :: Z=17 :: S=3
 :: GOTO 490
890 K=1 :: FOR I=13 TO 21 ST
EP 8 :: FOR Q=3 TO 18 :: DIS
PLAY AT(Q,I)SIZE(4):H$(ASC(S
EG$(HEX$,K,1))-47):: DISPLAY
 AT(Q,I+4)SIZE(4):H$(ASC(SEG
$(HEX$,K+1,1))-47)
900 K=K+2 :: NEXT Q :: NEXT
I :: RETURN
910 GOSUB 930
920 CALL KEY(1,K,S):: IF K=1
8 THEN Z=19 :: S=3 :: GOTO 4
90 ELSE 920
930 CALL HCHAR(20,15,32,16):
: Z1=21 :: S1=14 :: HEX$,HR$
="" :: HCV=1
940 FOR SF=0 TO 8 STEP 8 ::
FOR ZF=0 TO 8 STEP 8 :: FOR
A=0 TO 7 :: FOR Q=0 TO 4 STE
P 4 :: C2=0 :: FOR I=0 TO 3
:: CALL GCHAR(3+ZF+A,18+SF-I
+Q,C1)
950 C2=C2+2^I*(C1-40):: NEXT
 I :: S1=S1+1 :: HEX$=HEX$&C
HR$(48+C2)
960 IF C2>9 THEN HR$=HR$&CHR
$(C2+55)ELSE HR$=HR$&CHR$(C2
+48)
970 IF S1>30 THEN S1=15 :: Z
1=Z1+1
980 CALL SOUND(10,-5,0):: CA
LL HCHAR(Z1,S1,48+C2):: NEXT
 Q :: NEXT A :: NEXT ZF :: N
EXT SF :: CALL S2 :: CALL S2
 :: CALL S2 :: RETURN
990 GOSUB 1000 :: Z=21 :: GO
TO 490
1000 IF HCV=0 THEN CALL SOUN
D(3000,440,0):: CALL C :: DI
SPLAY AT(20,13):"WARNING\" :
: DISPLAY AT(22,13):"USE HEX
OUT BEVOR" :: GOTO 1090
1010 CALL CHAR(128,HR$):: CA
LL C :: CALL SOUND(200,880,9
,770,9,660,9):: DISPLAY AT(2
0,13):"' COLOR" :: DISPLAY A
T(22,13):"' MAGNIFY" :: DISP
LAY AT(24,13):"' END"
1020 Q=3 :: CALL MAGNIFY(3):
: CALL SPRITE(#25,128,2,160,
208)
1030 Z1=20 :: S1=15 :: GOSUB
 1160 :: IF X=4 THEN 1030
1040 IF Z=24 THEN 1090
1050 IF Z=22 THEN 1070 ELSE
ACCEPT AT(20,21)SIZE(2)VALID
ATE(DIGIT)BEEP:I$ :: IF I$="
" THEN 1050 ELSE CO=VAL(I$)
1060 IF CO<1 OR CO>16 THEN 1
050 ELSE CALL COLOR(#25,CO):
: GOTO 1030
1070 IF Q=3 THEN CALL MAGNIF
Y(4):: Q=4 ELSE CALL MAGNIFY
(3):: Q=3
1080 CALL HCHAR(22,15,96)::
GOTO 1030
1090 CALL DELSPRITE(ALL):: C
ALL SOUND(1,440,0):: S=3 ::
RETURN
1100 CALL C :: DISPLAY AT(21
,15):"HTAB\  VTAB\" :: HCV=0
 :: CALL MAGNIFY(2):: Z,S=1
:: CALL SPRITE(#9,91,14,7+Z*
8,102+S*8):: DISPLAY AT(23,1
7):"1        1"
1110 CALL KEY(1,K,I):: IF K<
>18 THEN 1130 ELSE CALL GCHA
R(Z+2,S+14,I)
1120 IF I=40 THEN CALL SOUND
(100,220,0,247,0,262,0):: CA
LL HCHAR(Z+2,S+14,41)ELSE CA
LL SOUND(99,440,0,494,0,524,
0):: CALL HCHAR(Z+2,S+14,40)
1130 CALL JOYST(1,X,Y):: IF
X=0 AND Y=0 THEN 1110
1140 Z=Z-Y/4 :: S=S+X/4 :: I
F Z>16 OR Z<1 THEN 510 ELSE
 IF S>16 OR S<1 THEN 510
```

```
115Ø CALL SOUND(3Ø,-2,Ø):: C
ALL LOCATE(#9,7+Z*8,1Ø2+S*8)
:: DISPLAY AT(23,15):USING "
###    ###":S,Z :: GOTO 111Ø
116Ø Z=Z1 :: S=S1 :: CALL S
:: CALL HCHAR(Z,S,1Ø4)
117Ø CALL KEY(1,K,I):: IF K=
18 THEN CALL HCHAR(Z,S,1Ø5):
: :: CALL S2 :: GOTO 123Ø
118Ø CALL JOYST(1,X,Y):: IF
X=4 THEN 122Ø
119Ø IF Y=Ø THEN 117Ø ELSE C
ALL HCHAR(Z,S,96)
12ØØ Z=Z-Y/4 :: IF Z>24 OR Z
<3 THEN 116Ø
121Ø CALL GCHAR(Z,S,FELD)::
IF FELD<>96 THEN 12ØØ ELSE C
ALL SOUND(1ØØ,-1,9):: CALL H
CHAR(Z,S,1Ø4):: GOTO 117Ø
122Ø CALL S :: CALL HCHAR(Z,
S,96)
123Ø RETURN
124Ø DATA CREATE\,,' WORKING
,,' OLD ONE,,OUTPUT\,,' HEX
CODE,,' LOOK,,,STORAGE\,,' S
AVE,,,,,,
125Ø DATA ' SYMMETRY,,' INVE
RT,,' ROTATION,,' MIRROR,,,'
 CLEAR,,' SET ALL,,,' HEX IN
,,' HEX OUT,,' LOOK,,' READ
Y,
126Ø FOR A=Ø TO 1 :: FOR I=Ø
 TO 7 :: DISPLAY AT(I+3+A*8,
13):SEG$(D$(A),I*16+1,16)::
NEXT I :: NEXT A :: RETURN
127Ø SUB SET :: CALL SOUND(-
999,-5,9):: CALL VCHAR(3,14,
39,16):: CALL HCHAR(19,14,38
):: CALL HCHAR(19,15,37,16)
128Ø FOR I=3 TO 18 :: CALL S
OUND(-1Ø,-5,Ø):: CALL HCHAR(
I,15,4Ø,16):: NEXT I :: CALL
 CHAR(96,"ØØ552A552A552A55",
1Ø4,"ØØØ8ØC7E7F7EØCØ8",1Ø5,"
ØØØØ1C3E3E3E1C"):: SUBEND
129Ø SUB C :: FOR I=2Ø TO 24
 :: DISPLAY AT(I,13):"" :: N
EXT I :: SUBEND
13ØØ SUB S2 :: FOR I=Ø TO 5
:: CALL SOUND(-2ØØ,11Ø*2^I,4
):: NEXT I :: SUBEND
131Ø SUB DIR :: FOR I=3 TO 2
4 :: DISPLAY AT(I,1)SIZE(11)
:"" :: NEXT I :: FOR I=3 TO
24 :: READ I$ :: IF I$="" TH
EN CALL SOUND(2Ø,-3,9)ELSE C
ALL SOUND(2Ø,-2,9)
132Ø DISPLAY AT(I,1)SIZE(11)
:I$ :: NEXT I :: SUBEND
133Ø SUB S :: CALL SOUND(-5Ø
,33Ø,4,22Ø,4):: CALL SOUND(-
5Ø,33Ø,4,44Ø,4):: SUBEND
134Ø !
135Ø ! *** PROGRAMMENDE ***
```

Fortsetzung von Seite 135

Hexin:

Hier haben Sie die Möglichkeit, den bereits existierenden Hexcode eines Sprites einzugeben. Ist dies geschehen, so können Sie an diesem Sprite nach Belieben Änderungen vornehmen, oder sich selbiges in „Natura“ anschauen. (Dieses Programm mit dem Namen „LOOK“ wird später erläutert!).

Dieses Programm ist sehr nützlich, wenn Sie sich die Grafiken anderer Listings einmal genauer ansehen wollen, bevor Sie das Listing eingeben, oder wenn Sie die Grafiken anderer Programme für eines Ihrer eigenen Produkte verwenden wollen.

Hex out:

Dieses Programm nimmt Ihnen die mühsamste Arbeit bei der Spritedefinierung ab. Es erzeugt den Hexcode des Sprites. Dieser wird unmittelbar unterhalb des Zeichenfeldes in vier Zeilen ausgegeben, wobei jede Zeile 16 Zeichen enthält und damit dem Hexcode eines Zeichens entspricht. (Ein Sprite besteht ja aus vier Zeichen!)

1. Zeile .... linkes oberes Viertel des Sprites
2. Zeile .... linkes, unteres Viertel des Sprites.
3. Zeile .... rechtes, oberes Viertel des Sprites
4. Zeile .... rechtes, unteres Viertel des Sprites

Wenn Sie den Hexcode in einem Programm verwerten wollen, dann geben sie ihn einfach der Reihe nach von Zeile 1–4 ein.

Bevor Sie die Programme wie „SAVE“ oder „LOOK“ verwenden, müssen Sie das HEX OUT-Programm verwenden, da der Computer den Hex-Code auch für diese Programme zum Rechnen verwendet.

Look

Mit der Hilfe dieses Programmes können Sie sich Ihre fertigen Sprites in Originalgröße anschauen. Sobald Sie dieses Programm aufgerufen haben, erscheint das Sprite am rechten, unteren Bildschirmrand, und daneben eine neues Menue. Hier können Sie die Farbe des Sprites mittels COLOR ändern, oder es mittels MAGNIFY vergrößern. Eine neuerliche Anwendung von Magnify bringt das Sprite wieder in die Ursprungsgröße zurück.

Bevor Sie dieses Programm verwenden können, müssen Sie das Hex-Out Programm durchlaufen haben. Haben Sie dies vergessen, so weist der Computer darauf hin.

Wenn Sie genug gesehen haben, steigen Sie mittels END aus und kehren so wieder in das Arbeitsmenue zurück.

Zum Schluß die drei einfachsten Teile des Arbeitsmenues:

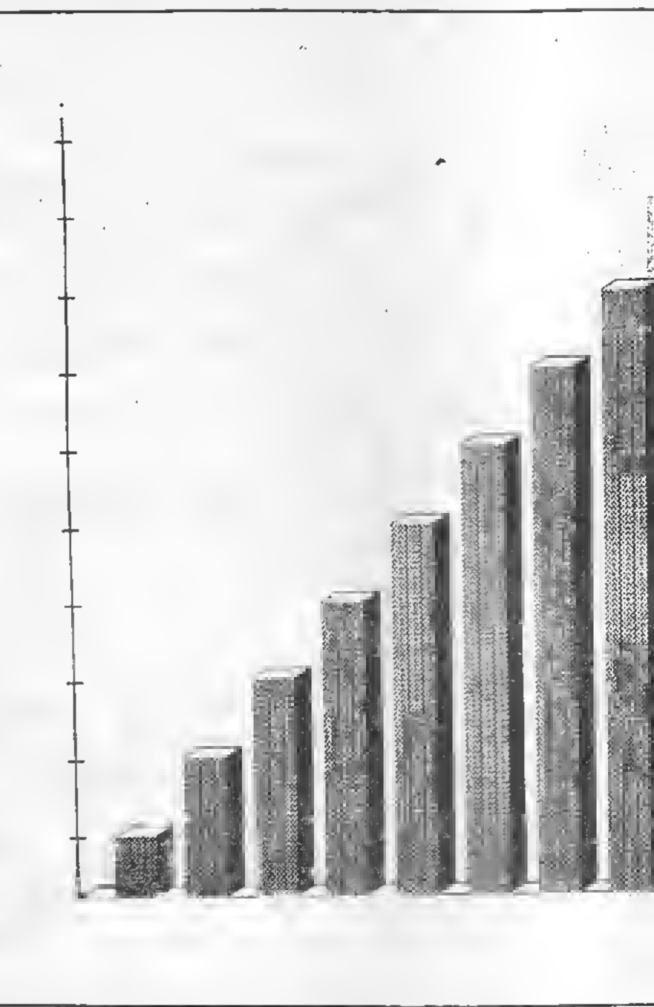
Clear

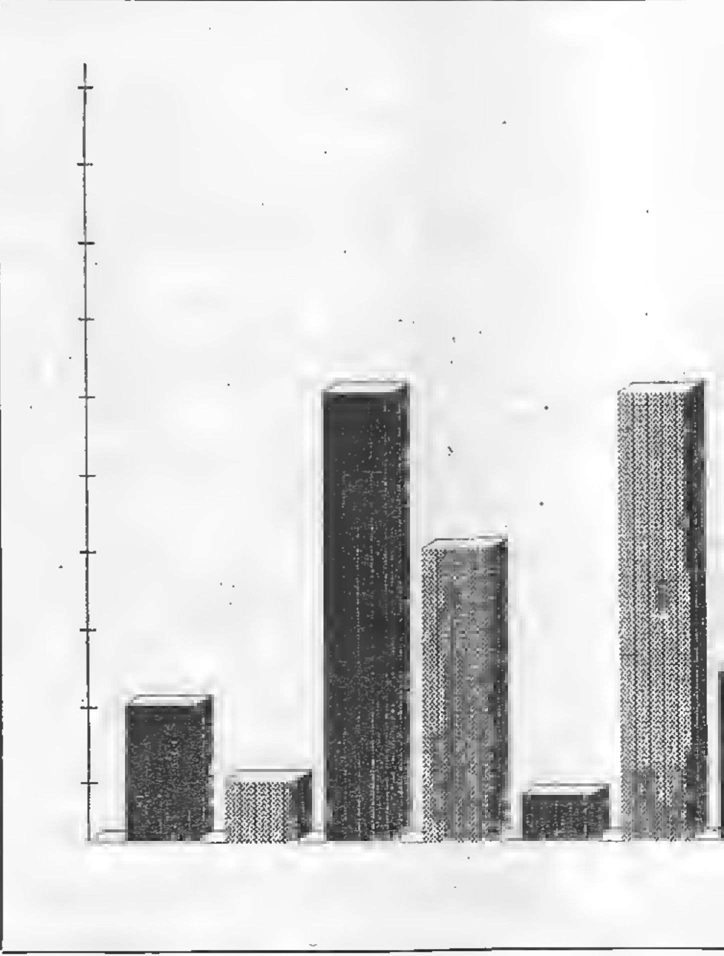
löscht das Zeichenfeld. Das dargestellte Sprite verschwindet.

Vorsicht: Sprites, die vorher gesaved wurden, sind nach der Anwendung von CLEAR nicht wieder aufrufbar. Sie sollten daher fertige Sprites, die Sie noch benötigen,

Bitte lesen Sie weiter auf Seite 146

WAS MAN SCHWARZ AUF WEISS BESITZT





Dies ist ein Programm, das dem creativen Anwender viele Variationsmöglichkeiten bietet, Säulen 3-dimensional mit dem Drucker RX-80 darzustellen.

Es ist für alle, die ihre Säulen schwarz auf weiß in der Hand halten wollen und auf größtmögliche Freiheit und Komfort bei der Gestaltung und Ausführung solcher Anwenderprogramme Wert legen, gedacht.

Dieses Programm ist in folgende Abschnitte unterteilt:

1. Eingabe: Hier werden alle wichtigen Voraussetzungen für den weiteren Ablauf von Ihnen eingegeben, wodurch die Variationen der verschiedenen Drucke erst zustande kommt. Sie können hier, im Dialog mit dem Computer, neben der Anzahl der Säulen auch entscheiden, ob die Y-Achse, ein Zwischenraum, oder im 2-Farben-Druck gedruckt werden soll.

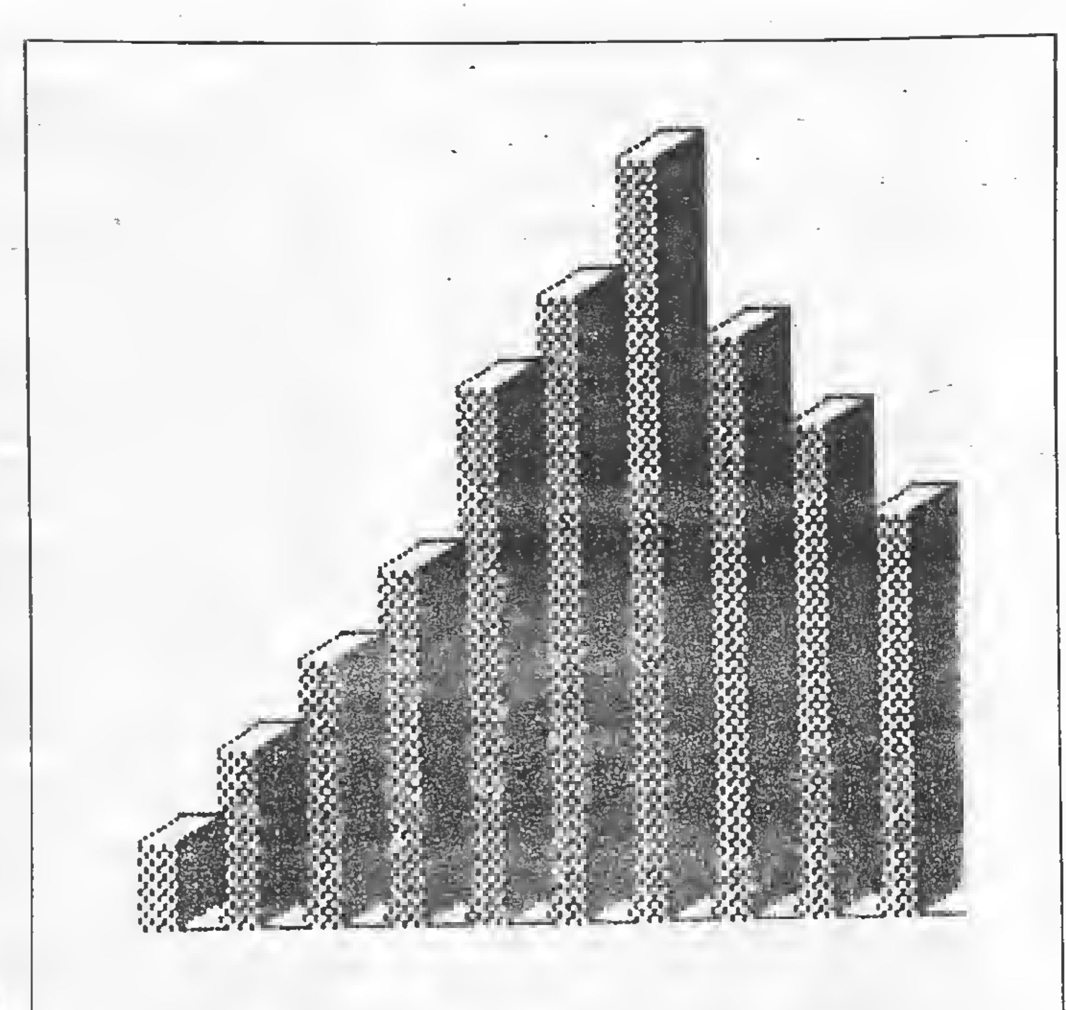
Zum 2-Farben-Druck ist anzumerken, daß hier in abwechselnder Folge 2 Säulen in verschiedenen Farben gedruckt werden, z.B. um einen Vergleich darzustellen. Eine weitere Möglichkeit, die Grafik ganz erheblich zu verändern, stellt die Eingabe der Säulenbreite dar.

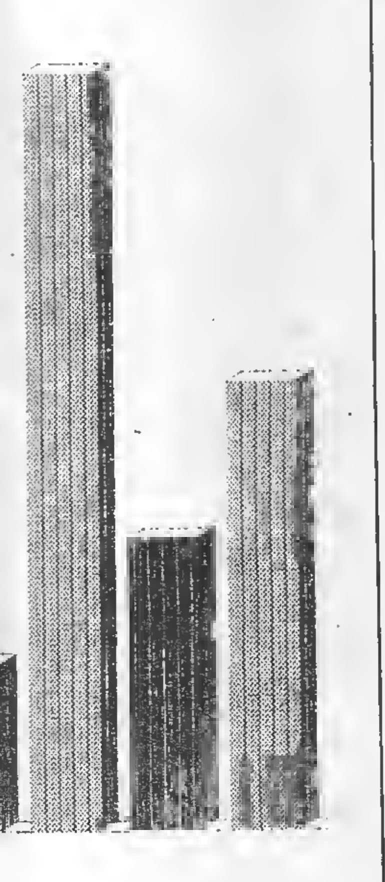
Mit der Eingabe des Faktors lassen Sie die Höhe der Säulen hochrechnen. Geben Sie z.B. den Faktor .5 ein, dann rechnet der Computer alle Werte auf 100 hoch.

2. Werteeingabe: Hier geben Sie die Werte für die Säulenhöhen ein. Angezeigt wird hier neben dem letzten von Ihnen eingegetippten Wert und der Anzahl der bisher eingetippten Säulen auch der höchste Wert, der nicht überschritten werden darf.

Michael Büth







Schematischer
Aufbau einer Säule
mit Verweis auf die Zeilennummern
wo die betreffenden Stücke entstehen
und gedruckt werden einschließlich
Kommentar

INT (A(I))
Z. – 1440–1460
2 Farben möglich:
schwarz Farb= 140
grau Farb= 139

Druckrichtung

Z.– 1520–1540

BREI 1/2
stellen Wert
für die Breite
d. Säule dar

Z.– 1560–1580
abhängig von BREI

Z.– 1630–1650

INT(A(I)
Z.– 1660
immer schwarz

Z.– 1670–1690

```
10 ! **********************
11 ! *                    *
12 ! *  SAEULENDIAGRAMM   *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *                    *
16 ! *   Michael  Bueth   *
17 ! *                    *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *    Epson  RX-80    *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     4279 Bytes     *
28 ! *                    *
29 ! **********************

30 !
400 DIM A(99)
410 OPEN #1:"PIO"
420 CALL CHAR(135,"0000FF")
430 CALL CLEAR :: CALL SCREE
N(14)
440 FOR UI=1 TO 13 :: CALL C
OLOR(UI,16,2)
450 NEXT UI
460 REM
470 REM ****************
480 REM *  EINGABE DER  *
490 REM *               *
500 REM *VORAUSSETZUNGEN*
510 REM ****************
520 REM
530 DISPLAY AT(2,7):"SAUELEN
DIAGRAMM"
540 CALL HCHAR(3,9,135,15)
550 DISPLAY AT(5,6):"von mic
hael bueth "
560 DISPLAY AT(10,2):"FAKTOR
 :"
570 ACCEPT AT(10,25)SIZE(9)V
ALIDATE(DIGIT,"."):V
580 DISPLAY AT(12,2):"Y-ACHS
E ? (J/N)"
590 ACCEPT AT(12,25)VALIDATE
("JN"):M$
600 IF M$="N" THEN GHJ=2
610 DISPLAY AT(14,2):"2-FARB
EN-DRUCK (J/N)"
620 ACCEPT AT(14,25)VALIDATE
("JN"):HN$
630 IF HN$="N" THEN 650
640 FDR=20
650 DISPLAY AT(16,2):"WIEVIE
LE SAUELEN ?"
660 ACCEPT AT(16,25)SIZE(2)V
ALIDATE(DIGIT,""):ANZ
670 IF ANZ>99 THEN 660
680 IF FDR=20 THEN 720
690 DISPLAY AT(18,2):"SAUELE
NBREITE :"
700 ACCEPT AT(18,25)SIZE(2):
BREI2
710 GOTO 760
720 DISPLAY AT(18,2):"ERSTE
SAUELENBREITE ?"
730 ACCEPT AT(18,25)SIZE(1)B
EEP VALIDATE(DIGIT,"12345"):
BREI1
740 DISPLAY AT(20,2):"ZWEITE
 SAUELENBREITE ?"
750 ACCEPT AT(20,25)SIZE(1)B
EEP VALIDATE(DIGIT,"12345"):
BREI2
760 CALL SOUND(100,110,0)
770 DISPLAY AT(22,2):"ZWISCH
ENRAUM ? (J/N)"
780 CALL KEY(0,CODE,STATUS)
790 IF STATUS=0 THEN 780 ::
IF CODE=74 THEN 810
800 ZWR=50
810 CALL KEY(0,CODE,STATUS)
820 IF STATUS=0 THEN 810
830 REM
840 REM ******************
850 REM **  WERTEINGABE  **
860 REM ******************
870 REM
880 CALL CLEAR
890 DISPLAY AT(2,7):"SAUELEN
DIAGRAMM"
900 LI=50/V
910 CALL HCHAR(3,9,135,15)
920 FOR I=1 TO ANZ
930 IF ASD>ANZ-1 THEN 1760
940 DISPLAY AT(7,2):"WERT :"
950 DISPLAY AT(14,6):"WERT D
ARF NICHT ";LI
960 CALL HCHAR(8,4,135,25)
970 DISPLAY AT(9,8):"bitte w
ert eingeben "
980 CALL SCREEN(2)
990 DISPLAY AT(16,7):"UEBERS
CHREITEN !"
1000 DISPLAY AT(7,20):"? "
1010 ACCEPT AT(7,15)SIZE(5)B
EEP VALIDATE(DIGIT,"12345678
90."):X
1020 DISPLAY AT(7,15):"---"
1030 DISPLAY AT(22,3):"LETZT
ER WERT : ";X
1040 X=X/1.0 :: X=X*V
1050 ASD=ASD+1 :: DISPLAY AT
(20,3):"SAEULEN      : ";ASD
1060 A(I)=X :: NEXT I :: CAL
L CLEAR :: HAI=1
1070 CALL SCREEN(9):: ST=0
1080 PRINT #1:CHR$(27);"1";
1090 PRINT #1:CHR$(27);"E";
1100 PRINT #1:CHR$(27);"m";C
HR$(4);
1110 REM
```

```
1120 REM ******************
1130 REM **    AUSDRUCK   **
1140 REM ******************
1150 REM
1160 FOR I=1 TO ANZ
1170 GHJ=GHJ+1 :: IF GHJ>1 T
HEN 1330
1180 RESTORE 1950
1190 PRINT #1:CHR$(27);"L";C
HR$(6);CHR$(0);
1200 FOR TGH=1 TO 6
1210 READ BRED :: PRINT #1:C
HR$(BRED);:: NEXT TGH
1220 FOR AN=1 TO 10
1230 ST=ST+1
1240 IF ST<2 THEN 1260
1250 PRINT #1:CHR$(133);
1260 PRINT #1:CHR$(133);CHR$
(133);CHR$(133);CHR$(128);
1270 NEXT AN
1280 PRINT #1:CHR$(133)
1290 RESTORE 1960
1300 PRINT #1:CHR$(27);"L";C
HR$(9);CHR$(0);
1310 FOR RW=1 TO 9 :: READ D
FV :: PRINT #1:CHR$(DFV);
1320 NEXT RW :: PRINT #1:CHR
$(160):: GOSUB 1720
1330 PRINT #1:CHR$(27);"3";C
HR$(20);
1340 IF ASD>1 THEN 1350 :: G
OSUB 1720
1350 IF FDR=20 THEN 1370
1360 GOTO 1410
1370 REM
1380 ASD=ASD+1
1390 IF ASD=2*INT(ASD/2)THEN
 FARB=139 ELSE FARB=140
1400 GOTO 1420
1410 FARB=139
1420 BG=0
1430 IF A(I)<2 THEN GOSUB 17
20 :: BG=BG+1 :: GOTO 1590
1440 FOR DS=1 TO INT(A(I))-1
1450 PRINT #1:CHR$(FARB);
1460 NEXT DS
1470 REM
1510 BG=BG+1 :: IF BG>1 THEN
 1560
1520 RESTORE 1910 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1530 FOR TG=1 TO 8 :: READ P
O :: PRINT #1:CHR$(PO);
1540 NEXT TG :: PRINT #1:CHR
$(160)
1550 GOTO 1590
1560 RESTORE 1920 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1570 FOR HG=1 TO 8 :: READ H
U :: PRINT #1:CHR$(HU);
1580 NEXT HG :: PRINT #1:CHR
$(160)
1590 IF FARB=139 THEN 1610
1600 IF BG>BREI1-1 THEN 1620
 :: GOTO 1430
1610 IF BG>BREI2-1 THEN 1620
 :: GOTO 1430
1620 IF A(I)<2 THEN 1700
1630 RESTORE 1930 :: PRINT #
1:CHR$(27);"L";CHR$(12);CHR$
(0);
1640 FOR MN=1 TO 12 :: READ
LOK :: PRINT #1:CHR$(LOK);
1650 NEXT MN
1660 FOR DFE=1 TO A(I)-2 ::
PRINT #1:CHR$(140);:: NEXT D
FE
1670 RESTORE 1900 :: PRINT #
1:CHR$(27);"L";CHR$(8);CHR$(
0);
1680 FOR AF=1 TO 8 :: READ Q
 :: PRINT #1:CHR$(Q);
1690 NEXT AF :: PRINT #1:CHR
$(160)
1700 IF ZWR=50 THEN 1710 ::
GOSUB 1720
1710 NEXT I :: GOTO 1760
1720 RESTORE 1940 :: PRINT #
1:CHR$(27);"L";CHR$(9);CHR$(
0);
1730 FOR UJ=1 TO 9 :: READ X
CV :: PRINT #1:CHR$(XCV);
1740 NEXT UJ :: PRINT #1:CHR
$(160)
1750 RETURN
1760 FOR GH=1000 TO 110 STEP
 -50 :: CALL SOUND(-10,GH,7,
GH+20,20):: NEXT GH
1770 DISPLAY AT(10,1):"KOPIE
 ERWUENSCHT ? (J/N) "
1780 ACCEPT AT(10,26)SIZE(1)
VALIDATE("JN"):K$
1790 IF K$="N" THEN 1820 ELS
E 1800
1800 ST=0
1810 IF M$="N" THEN GHJ=2 EL
SE GHJ=0
1815 GOTO 1070
1820 DISPLAY AT(15,1):"NEUE
WERTEEINGABE UNTER"
1830 DISPLAY AT(17,1):"GLEIC
HEN VORRAUSSETZUNGEN ?"
1840 ACCEPT AT(21,16)SIZE(1)
VALIDATE("JN"):C$
1850 IF C$="J" THEN 1870
1860 GOTO 1880
1870 ASD=0 :: IF M$="N" THEN
 GHJ=2 ELSE GHJ=0
1875 GOTO 880
1880 CLOSE #1
```

```
1890 CALL SCREEN(2):: CALL C
LEAR :: END
1900 DATA 255,127,63,31,15,7
,3,255
1910 DATA 128,64,32,16,8,4,2
,1
1920 DATA 0,0,0,0,0,0,0,255
1930 DATA 255,128,192,224,24
0,248,252,254,255,255,255,25
5
1940 DATA 255,0,0,0,0,0,0,0,
255
1950 DATA 31,24,20,18,17,16
1960 DATA 255,0,0,0,0,64,32,
16,15
```

TIPS & TRICKS

DER KLEINE MUNTERMACHER

Eine kleine Hardwareerweiterung macht den langsamen TI 99/4A zum Rennwagen.

Haben Sie das auch schon einmal erlebt:
Sie haben eine ganz tolle Idee für ein Programm und bemerken beim Programmieren, daß Ihr Vorhaben ein Flop werden muß, da die notwendigen Routinen den Rechner viel zu langsam machen? Und dann ärgern Sie sich darüber, weil die Assemblerausrüstung für Ihren TI viel zu teuer ist? Das ist nun vorbei.
Wir haben für Sie eine Hardwareschaltung zur Assemblerprogrammierung getestet, die zum einen sehr günstig zu kaufen ist und zum anderen nur das Extended Basic-Modul benötigt.
Damit Sie uns richtig verstehen: den Cassettenrecorder brauchen Sie auch noch, denn irgendwo müssen Ihre Programme ja gespeichert werden. Sie haben richtig gelesen: Es wird kein Diskettenlaufwerk benötigt. Und jetzt in medias res.

DER KASTEN

Die kleine Hardwareschaltung findet in einem Kunststoffkasten von 100 mm x 50 mm x 25 mm Platz und besteht, wenn Sie das Gehäuse geöffnet haben, aus sehr wenigen Bausteinen. Auf der Platine finden sich zwei Widerstände, ein Chip vom Typ HM6116P-3, ein Speicherbaustein, und ein SN74LS138N.
Ein kleiner Kippschalter wurde an der Gehäusestirnseite untergebracht. Der elektronische Aufbau birgt also weder Wunder noch technische Novation. Aber sinnreiche Details. Dieser kleine Kasten zeigt an beiden Längsseiten einen Peripheriebus. Mit dem einen Stecker können Sie damit Ihre Neuerwerbung an den Systembus Ihres TI anschließen und die andere Busseite für weitere Peripherie nutzen. Falls vorhanden. Die Stecker passen übrigens sauber an den Tl-Bus, da klemmt oder wakkelt nichts. Ebenfalls von sehr guter Qualität ist das Kunststoffgehäuse. Es besteht aus zwei Hälften, die sehr stabil aussehen und mit vier starken Schrauben zusammengehalten werden. Einen Sturz von der Tischplatte dürfte das Gehäuse wohl aushalten. Aber versuchen Sie bitte nicht, mit Ihren Füßen die Stabilitätsgrenze des Gehäuses zu ergründen. Die beiden Gehäusehälften werden nicht einfach stumpf aufeinander gesetzt, sondern zeigen Nut und Feder. Das zeugt von durchdachter Konstruktion. Derart verbundene Bauteile können nicht wackeln oder zueinander verrutschen. Nur die Farbgebung der Gehäuseteile läßt zu wünschen übrig. Grau in Grau ist die Devise. Oberteil in Hellgrau Unterteil in Dunkelgrau. Aber Ihr TI soll ja auch kein buntes Kunstwerk sein.

AUSSTATTUNG

Das kleine Kästchen kommt natürlich nicht alleine. In seiner Begleitung finden sich eine Kassette und ein kleines Handbuch. Klein ist etwas untertrieben, denn im Format DIN A4 läßt es sich schwer verlieren und mit 25 Seiten hat es schon recht viel zu lesen.
Im Handbuch wird nicht nur beschrieben, wie Sie Ihr kleines Zauberkästchen anschließen müssen. Sie werden in einem Kurzlehrgang in die Assemblerkürzel eingewiesen. Und zwar in alle, die auf dem TMS 9900 lauffähig sind. Und das Ganze geschieht in deutscher Sprache. Obendrein findet sich ein kleines Beispielprogramm, das Ihnen den ersten Start leicht machen soll. Sie kommen aber in keinem Fall darum herum, sich selbst mit anderer Literatur zur Assemblerprogrammierung zu beschäftigen. Und da gibt es mittlerweile einiges auf dem Büchermarkt. Einen entsprechenden Vermerk mit Literaturhinweis finden Sie ebenfalls im Handbuch. Ein, wie wir meinen, lobenswerter Zug.
Die mitgelieferte Kassette hat dann endlich alles, was Sie sich schon lange gewünscht haben:
- Line by Line-Assembler
- Dissassembler
- Kassettenladeroutine
- Kassettenspeicherroutine

ANWENDUNG

Um Ihre Erweiterung in Betrieb nehmen zu können, genügt es, sie an den Tl-Bus zu stecken. Dazu muß die Konsole ausgeschaltet sein. Der schon erwähnte Kippschalter soll in der linken Position stehen. Danach schalten Sie Ihr Gerät ein, rufen Ex-Basic auf und bringen den Kippschalter in die andere Position. Jetzt können Sie den Befehl CALL INIT eingeben. Nach Eingabe von NEW ist Ihr Rechner betriebsbereit. Wenn Sie jetzt den Line by Line-Assembler von Kassette laden und starten, können Sie sofort mit der Programmierung Ihrer Ideen beginnen. Auf Wunsch erfolgt ein Protokollausdruck über einen angeschlossenen Drucker. Der Assembler

ähnelt sehr stark dem vom Mini Mem, kann aber von der Geschwindigkeit nicht mithalten und hat leider auch einige Einschränkungen, die nicht verschwiegen werden:

- nur eine Zahl pro Data-Anweisung zulässig
- außer Indizierung keine Berechnung im Operandenfeld
- Programme sind auf eine bestimmte Adresse festgelegt
- keine vorgegebenen Definitionen
- das S-Symbol als Synonym für den Wert des Programmzählers ist nicht zulässig.

Mit diesen Einschränkungen läßt es sich aber ganz gut leben, zumal Sie mit diesem Assembler ein wirkungsvolles Instrument in der Hand haben, das es auch erlaubt, direkte Manipulationen im VDP vorzunehmen. Die gesamte Handhabung des Assemblers ist problemlos zu nennen und bereitet viel Spaß, wenn Sie sich erst einmal eingearbeitet haben.
Eine weitere wertvolle Hilfe stellt der Disassembler dar, mit dem Sie z.B.: fremde Maschinenprogramme untersuchen können.
Damit Ihre Programme nicht verloren gehen, können Sie alles auf Kassette abspeichern. Sie brauchen keine schwer verständlichen Befehle einzugeben, denn die Speicherroutine sucht sich die Namenstabelle, Programmstartadresse und Länge des Programms selber.
Genauso unkompliziert funktioniert das Wiedereinlesen Ihres Programms. Da kann wirklich nichts schief gehen. Dafür sorgt die Einleseroutine.

ERKLÄRUNGEN

Die auf Kassette mitgelieferten Programme befinden sich schon recht lange auf dem Markt. Daraus macht auch das Handbuch kein Geheimnis. Es handelt sich hierbei um Ex-Basic-Programme, die erstmalig 1983 veröffentlicht wurden und bis Dato nur mit der 32 KRAM liefen. Diese Erweiterung ist zwar überall käuflich zu erwerben, kostet aber zwischen 450 und 650 DM. Die hier vorgestellte Erweiterung beläuft sich auf nur 149 DM. Selbst das Mini Mem, bisher der günstigste Einstieg in die Welt der Maschinensprache, kann da nicht mithalten. Falls Sie überhaupt noch irgendwo ein solches Modul erwerben können.
Diese kleine Hardwareschaltung läßt uns aufatmen. Der TI ist noch lange nicht tot. Jedenfalls solange nicht, wie solch gute Hardware auf den Markt dringt und auch in großen Stückzahlen verfügbar ist. Die Produktion läuft auf vollen Touren und wer sich solch ein Gerät zulegen will, der sei an die Firma RADIX Bürotechnik in Hamburg verwiesen. Sie werden mit Sicherheit nicht enttäuscht sein. Denn diese Hardware stellt eine echte Alternative zur Speichererweiterung oder dem fast nicht mehr zu bekommenden Mini Mem dar, hat fast den gleichen Befehlssatz, ist billiger und sehr einfach zu bedienen.

Klaus Weidemann

BÖRSE

Hilfe! Die Austrittsgebühr beim Rex Soft Club beträgt 85 DM. Deswegen muß ich folgende Sachen verkaufen: Joysticks (35 DM), Joystickadapter (15 DM) + Othello-Modul (15 DM) + Statistik-Modul (20 DM). Tel. 07142/64321

Centronics-Interface (Extern) für TI 99/4A – 149,– DM. Tel. 07821/7570

---

6 aus 49 + 7 aus 38. 2 Systemprogr. (X-Basic) mit 28 Systemen. Kass. 30 DM + Nachnahme von P. Hielscher, Am Wall 22, 4401 Saerbeck

---

Suche: Floppy SS/SD – 150,–. o. DSDD 80 T. – 300 ggf. a. preisw. neu sowie Speechsynt. – 70,–. Tel. 07240/5300

Verkaufe TI Basic/Extended Basic Handbuch gegen Höchstgebot 4 Wo. n. Erscheinung. Vers. p. NN. Franz Tersluisen, Tel. 02851/7438 n. 17 Uhr

---

Salpha-Software und Module Donkey-Kont 60, Dig-Dug 45, Defender 35, Burgertime 50, Q-Bert 60, Re.to Pirate I.50, Parsec 20, Tombstone Ci. 20, Alpiner 30, Chisholm Tr. 20, Munch-Man 30, Ti-Invaders 20. Tel. 0208 23 299 tgl ab 18 Uhr

---

Kompl. TI-Station: Box, alle Karten, 2 Floppys, Drucker, Kass. Rec., Konsole mit integ. Grafic X-Basic, E/A-Modul, Adventure, Super Duper, Dsk. Manager, Speech-Synthesizer, 37er Color-TV, + Zub. 2999,– DM VB – Infos Tel.: 030/ 686 54 03

TI 99/4A in Orig.-verpackg. + X-Basic + 2 Joysticks + Rec-kabel + Literatur + PGMs (Lohn- und Einkommensteuer) 550,–. 040/2 70 30 78 Gert Adler (10.00 bis 15.00 Uhr)

BÖRSE
BÖRSE
BÖRSE
BÖRSE
BÖRSE
BÖRSE
BÖRSE

Verkaufe TI99/4A + Exbasicmod. (original) + Schachmod. + Literatur. VB 500 DM (auch Einzelverkauf) Tel. 08039/26 30

Konsole; Joysticks; Synthesizer; X-Basic; EA; MM; X-Box; Diskettenlaufw.; Akustikkop.; RS232; Module; haufenweise Spiele; Informationsmaterial; Adressen; Bücher; Schaltpläne; Listings; Wert weit über 5500 VB 4200; kaum benutzt 760 24 06 (München)

TI 99/4A, P-Box, RS232, Disk + Contr., 32k, P-Code + USCD-Pasc., RGB-Mod., Joyst, E/A + dt. Handb., PIO-Kabel + Datenverw.-Modul, 4 Spielmod., Recorderkabel, alles einwandfrei und original TI. + Recorder + 15 Disks + ...
Kompl. 3200,– DM VB. 040/601 48 83

Hallo TI-Freaks. Verkaufe TI 99 Netzteil (50,– DM). Recorderkabel (10,– DM) orig. TI 99/4A Schaltnetzteil zum Austausch (50,– DM), TI-Tastatur (50,– DM), Jockstik (20,– DM) Kassettenrecorder (40,– DM), TI-Modulator (80,–) TI-99/4A Netzteil (30,– DM)
Suche TI-Module. F. Haage, Rudolfstr. 9, 746 BL-Frommern 07433/35189

++ GOLF ++ (TI-99/4A + ExBasic) Meistern Sie die 18 Bilder!! Viel Geschicklichkeit und Augenmaß sind gefordert! TOP GARBGRAPHIK! Bis 10 Spieler! DISK nur 10,– DM Vorauskasse! B&A AHRENS, Am Hinkelstein 21, 6140 Bernsheim 1

Verk. TI99/4A + Box + Floppy + 27 Disks + Ext. Bas + 6 Mod. Parsec-Minus-Schach-Blacky. Adventure-Disk Manager + Kassetten + Ori.Lit + Data Becker + Hefte + Zubehör. VB 1100. H.v. Smolinski, Schillerstr. 96, 2080 Pinneberg 04101–27233 FarbMoni +450 DM



SPRITE-KONSTRUKTEUR

Fortsetzung von Seite 139

unbedingt vorher mittels SAVE (das ist ein Programm des Arbeitsmenues) „aufheben“.

Set all:

SET ALL setzt alle Punkte. Abgesehen davon, daß das Zeichenfeld nach der Anwendung von SET ALL nicht weiß, sondern schwarz ist, hat dieses Programm die gleiche Eigenschaft wie CLEAR. Es ist sehr nützlich, wenn Sie Sprites mit großen Dunkelflächen definieren wollen.

Ready:

Bei der Anwendung von READY, dem letzten Punkt des Arbeitsmenues, kehren Sie in das Hauptmenue zurück. Das von Ihnen gezeichnete Sprite wird dabei nicht gelöscht. Sie können also gegebenenfalls wieder ins Arbeitsmenue mittels WORKING zurückkehren und Veränderungen an der Zeichnung vornehmen.

Das Hauptmenue

Wie bereits erwähnt, dient das Hauptmenue dazu, erstellte Sprites zu verwalten. Wenn Sie Ihren Joystick nach rechts drücken, gelangen Sie nicht in den Arbeitsmodus. Zeichnen können Sie nämlich nur, wenn Sie sich im Arbeitsmenue befinden.

Save

Mit SAVE können Sie fertige Sprites abspeichern und damit für spätere Verwendung aufheben.

Dabei fragt Sie der Computer nach dem Namen des Sprites. Dieser darf maximal 16 Buchstaben betragen. Drücken Sie die ALPHA-LOCK Taste und geben Sie den Namen ein. Nach Drücken von ENTER sagt Ihnen der Computer, das wievielte Sprite Sie soeben gesaved haben. Nicht der Spritename, sondern die Spritenummer ist ausschlaggebend für das „Zurückholen“ mittels CREATE.

Wurden bereits 18 Sprites gesaved, so erscheint bei versuchtem Einstieg in das Programm SAVE die Fehlermeldung WARNING: MEMORY FULL. Achtung: Diese Fehlermeldung kommt in diesem Fall nicht vom Betriebssystem des TI-99/4A, sondern wird softwaremäßig simuliert. Tatsächlich haben Sie auch bei 18 gespeicherten Sprites noch etwa 500 Bytes freien Speicherraum zur Verfügung.

Create:

(Nur in der Version 2.6—85, sonst OLD ONE)

Mit CREATE können Sie alte Sprites, die Sie aufgehoben haben, wieder aufrufen. Der Computer fragt Sie nach der Spritenummer des Sprites. Geben Sie eine Nummer ein, die nicht existiert, so ignoriert der Computer die Eingabe. Bei der Eingabe einer gültigen Spritenummer stellt der Computer das aufgerufene Sprite im Zeichenfeld dar und schreibt den Spritenamen darunter. (Dieser dient in erster Linie zur Orientierung für Sie).

Wurde noch kein Sprite gesaved, ist der Einstieg in das CREATE-Programm nicht möglich!

Außerdem gibt es im Hauptmenue noch die Programme HEX OUT und LOOK. Sie sind im Hauptmenue nochmals angeführt, damit Sie bei Bedarf nicht dauern zwischen Haupt- und Arbeitsmenue wechseln müssen.

Die Funktion der beiden Programme ist im Hauptmenue genauso wie im Neben- oder Arbeitsmenue.

Erläuterungen, Programm technische Daten:

Das Unterbringen eines so umfangreichen Programmes in einen 16k-Speicher wurde erst durch die Verwendung zahlreicher Unterprogramme möglich. Dementsprechend oft tauchen daher im Programm die Wörtchen GOSUB oder CALL auf.

Ein weiterer Vorteil des Programmes besteht darin, daß auf komplizierte Mathematik verzichtet wurde. Bei jeder Funktion tastet der Computer den Bildschirm ab und holt seine Informationen unmittelbar vom Schirm, was sehr viel Speicherplatz spart.

Achtung: Da das Programm, wie bereits erwähnt, von der Mathematik weitgehend Abstand nimmt, bilden die Befehle CALL GCHAR und natürlich auch CALL HCHAR die Grundlage für die Funktion.

Bei der Eingabe des Programmes mittels Listing in den Computer sei daher darauf zu achten, stets die richtigen Variablen oder Zahlen in diesen beiden Befehlen einzusetzen. Abtippfehler können zu schweren Logikfehlern führen, welche auch sehr schwer zu eruieren sein dürften.